Discrete Diffusion Schrödinger Bridge Matching for Graph Transformation

Jun Hyeong Kim[*]
Department of Chemistry
KAIST
Daejeon, Republic of Korea
junhkim1226@kaist.ac.kr

Seonghwan Kim[*]
Department of Chemistry
KAIST
Daejeon, Republic of Korea
dmdtka00@kaist.ac.kr

Seokhyun Moon[*]
Department of Chemistry
KAIST
Daejeon, Republic of Korea
mshmjp@kaist.ac.kr

Hyeongwoo Kim[*]
Department of Chemistry
KAIST
Daejeon, Republic of Korea
novainco98@kaist.ac.kr

Jeheon Woo[*]
Department of Chemistry
KAIST
Daejeon, Republic of Korea
woojh@kaist.ac.kr

Woo Youn Kim[†]
Department of Chemistry
KAIST
Daejeon, Republic of Korea
wooyoun@kaist.ac.kr

Abstract

Transporting between arbitrary distributions is a fundamental goal in generative modeling. Recently proposed diffusion bridge models provide a potential solution, but they rely on a joint distribution that is difficult to obtain in practice. Furthermore, formulations based on continuous domains limit their applicability to discrete domains such as graphs. To overcome these limitations, we propose Discrete Diffusion Schrödinger Bridge Matching (DDSBM), a novel framework that utilizes continuous-time Markov chains to solve the SB problem in a high-dimensional discrete state space. Our approach extends Iterative Markovian Fitting to discrete domains, and we have proved its convergence to the SB. Furthermore, we adapt our framework for the graph transformation and show that our design choice of underlying dynamics characterized by independent modifications of nodes and edges can be interpreted as the entropy-regularized version of optimal transport with a cost function described by the graph edit distance. To demonstrate the effectiveness of our framework, we have applied DDSBM to molecular optimization in the field of chemistry. Experimental results demonstrate that DDSBM effectively optimizes molecules' property-of-interest with minimal graph transformation, successfully retaining other features.

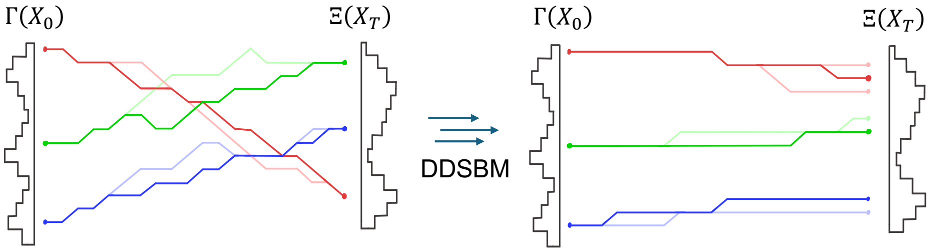
1 Introduction

Transporting an initial distribution to a target distribution is a foundational concept in modern generative modeling. Denoising diffusion models (DDMs) have been highly influential in this area, with a primary focus on generating data distributions from simple prior [1, 2, 3, 4, 5]. Despite their promising results, setting the initial distribution as a simple prior makes DDMs hard to work in tasks where the initial distribution becomes a data distribution, such as image-to-image translation. To tackle this, diffusion bridge models (DBMs) extend DDMs to transport data between arbitrary distributions [6, 7, 8]. However, training DBMs requires a coupling between the initial and target distributions, which is often difficult to obtain in practice. The Schrödinger Bridge (SB) provides an

[*] Equal contribution

[†] Corresponding author

AI for New Drug Modalities at NeurIPS 2024.



(a) Bridge process (b) Schrödinger Bridge

Figure 1: A schematic illustration of DDSBM transforming (a) bridge process to (b) Schrödinger Bridge in discrete spaces.

attractive framework for constructing a joint distribution of two data distributions while aligning with the underlying stochastic dynamics [9, 10, 11].

Formally, the SB problem seeks the stochastic process that connects two distributions and is closest to a reference process, measured by the Kullback-Leibler (KL) divergence [12]. The SB problem can be described as an entropy-regularized optimal transport (EOT) problem, which introduces an entropy term to the optimal transport (OT) objective, resulting in randomness in the transport process [13]. Here, the transportation cost is determined by the system's natural dynamics; for example, in the case of Brownian motion, the transportation cost becomes L^2 [14]. The SB/EOT can be computed efficiently using the Sinkhorn algorithm, though high-dimensional or large-scale data applications remain challenging [15, 16]. In recent, many methods have been proposed to approximate SB via distribution learning, utilizing techniques developed in DBM and DDM [14, 17, 18, 7, 19, 20, 21].

Despite the progress, most of the methods focus on the continuous spaces, where diffusion processes are represented by Brownian motion, and SB problems in discrete domains are less explored. It is particularly significant in fields that handle discrete state data, such as graphs or natural language [22, 23]. Directly applying frameworks for approximating SB formulated in continuous spaces to these domains limits its potential since it does not reflect the intrinsic properties of discrete data space. To bridge the gap, we propose a novel framework called Discrete Diffusion Schrödinger Bridge Matching (DDSBM) utilizing the continuous-time Markov chains (CTMCs) to solve the SB problem in a high-dimensional discrete setting. Our approach leverages Iterative Markovian Fitting (IMF), which was originally proposed for the SB problem in a continuous domain [19, 20].

We then extend our formulation to the graph domain, where the underlying process can be interpreted as independent modifications to both the nodes and edges [23]. In this case, the cost function of the corresponding EOT can then be regarded as the graph edit distance (GED), which is especially suited for systems where preserving graph similarity is crucial [24]. The molecular optimization in drug/material discovery is such a case in that molecules are represented as graphs. In addition, the goal is to obtain the molecules with desired molecular properties while retaining favorable properties in acclaimed molecules. Since molecular structures are closely related to their properties, it is highly advantageous to minimize structural changes (or graph editing) during the optimization.

To validate our framework, we evaluated the performance of DDSBM on molecular optimization tasks, with criteria of demonstrating optimal structural modifications to achieve desired property. DDSBM shows promising results in molecular distribution shift with minimum structural change compared to the previous graph-to-graph translation models. As a direct result of this, DDSBM retains multiple properties of the initial molecules, along with desired property. Lastly, we applied DDSBM for a more challenging task, where proper joint pairing between two molecular spaces can not be defined properly.

Our contributions are as follows:

- We propose a novel framework, DDSBM, for the SB problem in discrete state spaces by exploiting the IMF procedure and prove its convergence to the SB solution.
- We extend our framework to the graph domain, demonstrating a connection between the objective function and the GED.

- By reformulating molecular optimization as the SB problem, we show that our approach successfully addresses molecular optimization tasks.

2 Theoretical Background

Notations. We consider the *sample space* $\Omega = D([0, \tau], \mathcal{X})$, which is set of all left limited and right continuous (càdlàg) paths over the finite metric space $(\mathcal{X}, d_{\mathcal{X}})$. The space of *path measures* is denoted by $\mathcal{P}(\Omega) = \mathcal{P}(D([0, \tau], \mathcal{X}))$. The subset of *Markov* measures of which the transition probability are continuous and differentiable for time is denoted by $\mathcal{M}$. The *transition probability* from state x at time s to state y at time t will be represented by $P_{s:t}(x, y)$. Similarly, the *transition rate* or *generator* of a Markov measure as $A_s(x, y)$. For any $M \in \mathcal{M}$, the *reciprocal class* of M is denoted by $\mathcal{R}(M)$ (see Definition 2.1). For any $\mathbb{P} \in \mathcal{P}(\Omega)$, $\mathbb{P}_t$ denotes the marginal distribution at time t, $\mathbb{P}_{s,t}$ as the joint distribution at time s and t, $\mathbb{P}_{t|s}$ the conditional distribution at time t given state at time s. $X = (X_t)_{t\in[0,\tau]}$ denotes the canonical process given by $X_t(\omega) = \omega_t$ for $\omega = (\omega_s)_{s\in[0,\tau]} \in \Omega$. The reference measure $\mathbb{Q} \in \mathcal{M}$ is an irreducible Markov measure on Ω with its associated canonical filtration. We will denote the transition rate (or probability) of any Markov measures M other than $\mathbb{Q}$, using a superscript notation.

2.1 Schrödinger Bridge Problem

Schrödinger Bridge (SB) problem is finding a stochastic process that most closely resembles a given reference process, with a condition that initial and final marginal distributions are fixed as Γ and Ξ. Specifically, the reference process with the law $\mathbb{Q}$ is given, the optimality is achieved by minimizing Kullback-Leibler (KL) divergence to the reference process, $D_{\text{KL}}(\mathbb{P}\|\mathbb{Q})$. Thus the definition of the SB solution is as below:

$$\mathbb{P}^{\text{SB}} = \arg\min_{\mathbb{P}}\{D_{\text{KL}}(\mathbb{P}\|\mathbb{Q}) : \mathbb{P}_0 = \Gamma, \mathbb{P}_\tau = \Xi\}. \tag{1}$$

If we additionally fix the initial and terminal coupling $\mathbb{P}_{0,\tau}$, the optimality can be found easily as a mixture of bridges $\mathbb{P}_{0,\tau}\mathbb{Q}_{\cdot|0,\tau}$, which implies that finding the SB solution is equivalent to finding optimal coupling $\mathbb{P}^{\text{SB}}_{0,\tau}$ [13]. In particular, such optimal coupling is called static SB solution, which could be defined as follows:

$$\mathbb{P}^{\text{SB}}_{0,\tau} = \arg\min_{\mathbb{P}_{0,\tau}}\{D_{\text{KL}}(\mathbb{P}_{0,\tau}\|\mathbb{Q}_{0,\tau}) : \mathbb{P}_0 = \Gamma, \mathbb{P}_\tau = \Xi\}. \tag{2}$$

Note that the KL-divergence is decomposed into the entropy term $H(\mathbb{P}_{0,\tau})$ and the cross-entropy term $\mathbb{E}_{\mathbb{P}_{0,\tau}}[-\log q(x_\tau|x_0)]$, where q denotes the density function of $\mathbb{Q}$. The cross-entropy term is represented as L^2 distance when the $\mathbb{Q}$ is associated with the reversible Brownian motion. In general, the static SB problem is equivalent to the entropy-regularized optimal transport (EOT) problem with the cost function $c(x, y) = -\log q_{\tau|0}(y|x)$ [13]. While previous approaches have primarily focused on solving SB problems in continuous spaces, typically involving Brownian motion, our work differs by addressing SB solutions in discrete spaces, specifically dealing with cádlág paths. Despite this distinction, the core idea remains rooted in the intrinsic properties of the SB solution (see Theorem A.1).

2.2 Solution Method

To solve the SB problem, we adopted the Iterative Markovian Fitting (IMF) method, a technique that has been proposed in various studies [19, 20]. These works primarily address diffusion processes in Euclidean space, whereas our approach differs slightly as we focus on a continuous-time Markov chain over finite state space. Applying IMF to càdlàg paths in discrete spaces requires additional theoretical developments or modifications to ensure proper convergence and stability in this new context, which is described in Appendix A.

The SB solution is a mixture of pinned-down measures of $\mathbb{Q}(\cdot|X_0 = x_0, X_\tau = x_\tau)$, where the pair (x_0, x_τ) is drawn from the coupling $\mathbb{P}^{\text{SB}}_{0,\tau}$. Based on this, the projection method first constructs a reciprocal measure, which is mixture of pinned-down processes from a given initial coupling. Although each pinned-down process is Markov, the mixture generally loses the Markov property in general as collection of Markov process is non-convex [25]. Thus, it then identifies the Markov measure that is closest to the mixture. This yields an improved coupling, and the process is iterated to obtain a measure that is both the mixture of pinned-down measures and Markov– the desired SB solution.

In this context, we define the reciprocal projection to describe the construction of a reciprocal mixture from a given coupling (see Definition 2.1). Similarly, the term Markov projection is used to describe the approximation of a reciprocal process with a Markov process (see Definition 2.2).

Definition 2.1. *(Reciprocal Projection)*
$\Lambda \in \mathcal{P}(\Omega)$ is in the reciprocal class $\mathcal{R}(\mathbb{Q})$ with respect to a Markov measure $\mathbb{Q}$ if $\Lambda = \Lambda_{0,\tau}\mathbb{Q}_{|0,\tau}$. For a measure $\Lambda \in \mathcal{P}(\Omega)$, its reciprocal projection with respect to $\mathbb{Q}$ is

$$\Pi_{\mathcal{R}(\mathbb{Q})}(\Lambda)(\cdot) := \iint_{(\cdot)} \Lambda(dx_0, dx_\tau)\mathbb{Q}(dx_t|x_0, x_\tau).$$

Among the measures with the coupling $\Lambda_{0,\tau} \neq \mathbb{P}^{\text{SB}}_{0,\tau}$, the minimizer of the KL-divergence to the reference process is the (non-Markov) reciprocal projection $\Pi_{\mathcal{R}}(\Lambda)$. The reciprocal projection consists of a mixture of bridges, where each bridge is derived from Doob's h-transform with the realization of an end-point pair $(x, y) \sim \Lambda_{0,\tau}$ [26]. Obviously, it preserves the initial-terminal coupling. Although each pinned-down bridge is Markov (see Lemma A.4), the mixture is generally not Markov.

Definition 2.2. *(Markov Projection)*
Given a path measure $\Lambda \in \mathcal{R}(\mathbb{Q})$, a Markov path measure that minimizes the reverse KL-divergence to Λ is called as Markov projection of Λ,

$$\Pi_{\mathcal{M}}(\Lambda) = \arg\min_{M} \left\{ D_{KL}(\Lambda \| M) : M \in \mathcal{M} \right\}.$$

The Markov projection preserves the marginal distribution at all times t, but does not preserve the coupling. Furthermore, the generator and the reverse KL-divergence of the Markov projection over cádlág paths are explicitly derived in the Proposition A.2.

For a given reciprocal process $\Lambda^{(0)} \in \mathcal{R}(\mathbb{Q})$, we consider a sequence $(\Lambda^{(n)})_{n\in\mathbb{N}}$ which is defined by the recurrence relation:

$$\begin{aligned} \Lambda^{(2n+1)} &= \Pi_{\mathcal{M}}(\Lambda^{(2n)}), \\ \Lambda^{(2n+2)} &= \Pi_{\mathcal{R}(\mathbb{Q})}(\Lambda^{(2n+1)}), \end{aligned} \tag{3}$$

where $\Lambda^{(0)}_0 = \Gamma$ and $\Lambda^{(0)}_\tau = \Xi$. Under mild assumptions, the resulting sequence of measures of iterative projection converges to $\mathbb{P}^{\text{SB}}$ in law (see Theorem 2.3 and Appendix A.5).

Theorem 2.3. *(Convergence of Iteration)*
Assume that $D_{KL}(\Lambda^{(0)}_{0,\tau} \| \mathbb{P}^{SB}_{0,\tau}) < \infty$, $\Lambda^{(n)} \ll \mathbb{P}^{SB}$ for all $n \in \mathbb{N}$. Then the sequence of KL-divergence to $\mathbb{P}^{SB}$ non-increasing,

$$D_{KL}(\Lambda^{(2n)} \| \mathbb{P}^{SB}) \geq D_{KL}(\Lambda^{(2n+1)} \| \mathbb{P}^{SB}) \geq D_{KL}(\Lambda^{(2n+2)} \| \mathbb{P}^{SB}).$$

$D_{KL}(\Lambda^{(2n)} \| \mathbb{P}^{SB}) = D_{KL}(\Lambda^{(2n+1)} \| \mathbb{P}^{SB})$ if and only if $\Lambda^{(2n)} = \Lambda^{(2n+1)} = \mathbb{P}^{SB}$. Moreover, $\Lambda^{(n)}$ converges to $\mathbb{P}^{SB}$ in law as $n \to \infty$.

3 Methods

Here, we propose the Discrete Diffusion Schrödinger Bridge Matching (DDSBM) framework, which focuses on solving the SB problem in discrete state spaces. Approaches to the SB problem in continuous spaces model probability flows through stochastic differential equations, while our method uses continuous-time Markov chains (CTMCs) in discrete state spaces, governed by the Kolmogorov equation Equation (9). We first provide the algorithm of DDSBM (Section 3.1) to apply Iterative Markovian Fitting (IMF) to càdlàg paths in discrete spaces, of which the convergence is ensured theoretically by Theorem 2.3. We then discuss how DDSBM can be implemented in the graph transformation (Section 3.2) and introduce a graph permutation matching algorithm for efficiency (Section 3.3).

3.1 Algorithm for Solving the Schrödinger Bridge Problem

In this section, we present the framework for solving the SB problem in discrete state spaces based on the IMF algorithm proposed by Shi et al. [20]. We refer to it as the DDSBM framework. The framework iteratively applies Markov and reciprocal projections to update the sequence of measures, as described in Equation (3).

It begins with an initial random coupling π such that $\pi_0 = \Gamma$ and $\pi_\tau = \Xi$. Following the definition of the reciprocal class in Definition 2.1, we construct the initial reciprocal bridge to obtain the measure $\Lambda^{(0)}$.

Given a reciprocal measure $\Lambda^{(2n)} \in \mathcal{R}(\mathbb{Q})$, the next step is to compute its Markov projection $M^{(2n+1)} := \Pi_{\mathcal{M}}(\Lambda^{(2n)})$. The exact form of the transition rate for $M^{(2n+1)}$ is provided in Proposition A.2. In practice, the transition rate is approximated by a neural network.

To achieve this, it first samples pairs (x_0, x_τ) from $\Lambda^{(2n)}_{0,\tau}$ and samples intermediate states x_t by constructing the bridge $\mathbb{Q}(\cdot|X_0, X_\tau)$. The sampled pairs (x_t, x_τ) are distributed according to $\Lambda^{(2n)}_{t,\tau}$. Using these realizations, the rate matrix of $M^{(2n+1)}$ is approximated by parameterized Markov measure M^θ, by minimizing the following loss function:

$$\mathcal{L}(\theta) = \int_0^\tau \mathbb{E}_{\Lambda^{(2n)}_{t,\tau}} \left[(A_t^{\mathbb{Q}_{\cdot|\tau}} - A_t^{M^\theta})(X_t, X_t) + \sum_{y \neq X_t} A_t^{\mathbb{Q}_{\cdot|\tau}} \log \frac{A_t^{\mathbb{Q}_{\cdot|\tau}}}{A_t^{M^\theta}}(X_t, y) \right] dt, \qquad (4)$$

where $A_t^{\mathbb{Q}_{\cdot|\tau}}$ denotes the generator of pinned down process of $\mathbb{Q}(\cdot|X_\tau)$. From the approximated generator $A_t^{M^\theta}$, we can sample x_τ given x_0, where $(x_0, x_\tau) \sim M^\theta_{0,\tau} \approx M^{(2n+1)}_{0,\tau}$. Note that, until the sequence of measures converges, the new joint coupling $M^\theta_{0,\tau} \approx M^{(2n+1)}_{0,\tau}$ will differ from the previous one $\Lambda^{(2n)}_{0,\tau}$.

Once the Markov measure $M^{(2n+1)}$ is obtained, we proceed to compute the corresponding reciprocal measure $\Lambda^{(2n+1)}$ through reciprocal projection, $\Lambda^{(2n+1)} := \Pi_{\mathcal{R}(\mathbb{Q})}(M^{(2n+1)})$. In theory, as shown in Proposition A.2, the time marginal distributions are preserved under Markov projection, meaning that $\Lambda^{(2n)}_t = M^{(2n+1)}_t$ for all $t \in [0, \tau]$. However, in practice, since the Markov projection is approximated using a neural network, repeatedly applying this approximation can lead to an accumulation of errors in the time marginals. Such accumulated errors may violate the marginal condition of the SB problem, particularly leading to a potential failure in satisfying the terminal condition $\mathbb{P}_\tau = \Xi$.

To compensate these errors, the next Markov measure $M^{(2n+2)} := \Pi_{\mathcal{M}}(\Lambda^{(2n+1)})$ is approximated in a time-reversal way (see Proposition A.8). Based on the time-symmetric nature of Markov measures, we can leverage the time-reversed generator $\tilde{A}_t^{M^{(2n+2)}}$, which enables the sampling of x_0 conditioned on x_τ, where $(x_0, x_\tau) \sim M^{(2n+2)}_{0,\tau}$. The approximation of $\tilde{A}_t^{M^{(2n+2)}}$ is achieved by minimizing the following loss function:

$$\mathcal{L}(\phi) = \int_0^\tau \mathbb{E}_{\Lambda_{0,t}} \left[(\tilde{A}_t^{\mathbb{Q}_{\cdot|0}} - \tilde{A}_t^{M^\phi})(X_t, X_t) + \sum_{y \neq X_t} \tilde{A}_t^{\mathbb{Q}_{\cdot|0}} \log \frac{\tilde{A}_t^{\mathbb{Q}_{\cdot|0}}}{\tilde{A}_t^{M^\phi}}(X_t, y) \right] dt, \qquad (5)$$

where $\tilde{A}_t^{\mathbb{Q}_{\cdot|0}}$ denotes the time-reversal generator of pinned down process of $\mathbb{Q}(\cdot|X_0)$, and ϕ represents the parameters of the neural network approximating the time-reversed generator $\tilde{A}_t^{M^\phi}$.

In this manner, the iterative Markov projection following the reciprocal projection is performed alternately in a forward and backward (time-reversal) fashion. Finally, this process yields a sequence of measures $(\Lambda^{(n)})_{n\in\mathbb{N}}$ and $(M^{(n)})_{n\in\mathbb{N}^+}$, which converges to $\mathbb{P}^{\mathrm{SB}}$ in theory.

3.2 Process on the Graph

We present a method for applying the previously described solution to a graph transformation. A graph is represented as $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = (v^{(i)})_i$ and $\mathbf{E} = (e^{(ij)})_{ij}$ denote node and edge features, respectively. In a molecular graph, for example, the node and edge features correspond to atomic types and edge features, respectively. Here, $\mathbf{V}$ and $\mathbf{E}$ are modeled as products of categorical random variables.

As the reference process, we define a jump process in which the nodes and edges of the graph change discretely, assuming that all nodes and edges are independent. Therefore, the transition probability P and the rate A of the reference process is described as follows:

$$\begin{aligned} P^{\mathbf{G}}_{s:t}(\mathbf{G}_1, \mathbf{G}_2) &= \prod_i P^V_{s:t}(v_1^{(i)}, v_2^{(i)}) \prod_{i,j} P^E_{s:t}(e_1^{(ij)}, e_2^{(ij)}), \\ \partial_t P^{(\cdot)}_{s:t}(x, y) &= \sum_{z \in \mathcal{X}^{(\cdot)}} P^{(\cdot)}_{s:t}(x, z) A^{(\cdot)}_t(z, y), \\ A^{(\cdot)}_s(x, y) &= \partial_t P^{(\cdot)}_{s:t}(x, y)\big|_{t=s}, \end{aligned} \tag{6}$$

where $\cdot$ denotes V or E, $\mathbf{G}_1$ denotes $(\mathbf{V}_1, \mathbf{E}_1) = \Big((v_1^{(i)}), (e_1^{(ij)})\Big)$, $\mathbf{G}_2$ denotes $(\mathbf{V}_2, \mathbf{E}_2) = \Big((v_2^{(i)}), (e_2^{(ij)})\Big)$, and $\mathcal{X}^V$ and $\mathcal{X}^E$ denote the state space of the nodes and edges, respectively. More specifically, we use a monotonically decreasing function for signal to noise ratio, $\bar{\alpha} : [0, \tau] \to (0, 1]$, in which the transition rate is defined as:

$$A^{(\cdot)}_t(x, y) = \partial_t(\ln \bar{\alpha}(t)) \left(\delta_{xy} - \mathbf{m}^{(\cdot)}(y)\right), \tag{7}$$

where δ_{xy} denotes the Kronecker delta, and $\mathbf{m}^{(V)}$ and $\mathbf{m}^{(E)}$ denote the prior distribution of nodes and edges as proposed in the previous discrete diffusion work [23]. According to the Kolmogorov equation, we get the transition probability as,

$$P^{(\cdot)}_{s:t}(x, y) = \frac{\bar{\alpha}(t)}{\bar{\alpha}(s)} \delta_{xy} + \left(1 - \frac{\bar{\alpha}(t)}{\bar{\alpha}(s)}\right) \mathbf{m}^{(\cdot)}(y). \tag{8}$$

Note that the choice of $\mathbf{m}(\cdot)$ as uniform is associated to the diffusion on the state space $\mathcal{X}$, where the $\mathcal{X}$ is considered fully-connected graph. Moreover, the stationary distribution of the associated generator always becomes $\mathbf{m}(\cdot)$. Although non-uniform choice of $\mathbf{m}(\cdot)$ breaks the diffusion formulation on $\mathcal{X}$, it does not harm SB formulation.

3.3 Graph Permutation Matching

The transition probability of the reference process depends on graph permutations (see Equation (6)), so graph permutation matching must be considered beforehand. Although this issue does not affect the sampling phase, it becomes problematic when computing the likelihood of the reference process for two given graphs $\mathcal{G}$ and $\mathcal{G}'$, or when constructing a reciprocal process, which is a mixture of Markov bridges between the two graphs (see Appendix B).

One way to handle this is selecting a graph permutation that maximizes the likelihood under the reference process. Finding the optimal permutation can be formulated as a quadratic assignment problem (QAP), where the objective is to minimize the negative log-likelihood (NLL), consisting of the sum of the NLLs for both the nodes and edges. While exact solutions are possible through mixed integer programming, the problem is NP-hard, so alternative methods are preferred. Specifically, we employ a max-pooling algorithm by [27], which is an approximation method categorized by continuous relaxation. After obtaining an approximate solution, we use the Hungarian algorithm to discretize the assignment vector to the final solution. We observed that the graph matching is highly accurate in molecular graph matching (see Appendix B.5). The details of the algorithm are described in Appendix B.4. We utilized the graph matching algorithm for every reciprocal bridge construction and likelihood computation.

Recall that the SB problem could be interpreted to the EOT problem, where the cost function corresponds to the NLL. Thus, defining the reference process can be interpreted as defining a distance (cost) on the set of graphs. Interestingly, we found that the likelihood of optimal permutation is interpreted as the graph edit distance (see Appendix B.6). This leads to the conclusion that the SB problem, with the $\mathbb{Q}$ as Equation (8), is analogous to an OT problem over the metric space of graphs equipped with graph edit distance as metric (GED), where the GED computation is well known to be NP-hard problem.

4 Related Works

Diffusion Bridge Models. Diffusion bridge models (DBMs) have recently shown state-of-the-art results in a variety of continuous domains, such as images, biology, and chemistry [6, 7, 18, 8, 21]. While Igashov et al. [28], Yang et al. [29] have extended these models to discrete domains, they focused on settings where well-defined data pairs exist. To the best of our knowledge, we are the first to study DBM in discrete domains where proper joint distributions between data points are not provided or well-defined.

Schrödinger Bridge Problem. The Schrödinger Bridge (SB) problem is an important concept in recent generative modeling [17, 18, 19, 20]. In particular, incorporating the SB problem into DBM can address scenarios where no appropriate joint distribution is available, as demonstrated by recent works [9, 10, 11]. For example, Somnath et al. [18] proposed learning an SB based on an assumed (partial) true coupling, while Shi et al. [20] showed that it is possible to generate high-quality samples from arbitrary couplings that are well-aligned with the initial data. However, most existing approaches focus on continuous spaces. To the best of our knowledge, we are the first to propose a framework for solving the SB problem in high-dimensional discrete spaces.

Molecular Optimization. Molecular optimization is a promising strategy in drug/material discovery that aims to improve acclaimed molecules to satisfy multiple domain-specific properties. One major approach to the molecular optimization problem is to formulate it as a graph transformation problem, which can be categorized into latent-based and graph-editing approach. The latent based approaches such as JT-VAE by Jin et al. [30] and HierG2G by Jin et al. [31] encode an input graph into a single latent vector and then decode it into a whole graph that follows a certain data distribution. The graph-editing approaches such as Modof by Chen et al. [32] and DeepBioisostere by Kim et al. [33] learn a graph editing procedure to transport a given molecule to another. Despite their promising results in optimizing molecular graphs, their training schemes rely on paired data created by predefined rules, which limits not only the general applicability but also the optimality of the transformations. In this work, we propose a more flexible framework for molecular optimization by formulating it as a graph SB problem, leading to more optimal graph transformation accompanying less structural changes.

5 Results and Discussions

Here, we demonstrate the effectiveness of the Discrete Diffusion Schrödinger Bridge Matching (DDSBM) framework on graph transformation tasks. Specifically, we apply DDSBM to a chemical domain, where the graph transformation task is nontrivial due to additional constraints imposed by molecular graphs and their associated properties. We conducted experiments on two different molecule datasets: ZINC250K [34] and Polymer [35]. Throughout this section, we first elaborate on the common experimental setups and metrics for evaluation. The second and third sections provide a detailed analysis of ZINC250K and Polymer experiments.

5.1 Experimental setup and metrics

Experimental Setup. To train the models on graph transformation problems, an initial coupling between two distributions is necessary. We randomly coupled the data of initial and terminal distributions to obtain paired data. The molecular pairs are divided into training and test datasets in a ratio of 8:2. The effect of different initial couplings is discussed in Appendix D.3. Detailed explanations about model architectures and hyperparameters can be found in Appendix C.

Metrics. By definition of SB (see Equation (1)), both joint distribution and marginal distributions at each side must be examined to assess the degree to which the SB has been successfully achieved. Specifically, we evaluate these distributions from two perspectives: graph structure and molecular properties. While graph structure provides valuable insights, it alone may not be sufficient for molecular optimization tasks. Therefore, the assessment of molecular properties is essential for a more comprehensive validation of the models, as these properties might exhibit weak correlations with the graph structures.

In the context of graph structure, we calculate Fréchet ChemNet Distance (FCD) score [36] between the target molecule distribution and the generated molecules from each model, which focuses on the marginal distribution of the molecules. As the metric for evaluating the quality of the joint

Table 1: **Distribution shift performance on ZINC.** Reference refers to metrics from the initial coupling, used as a standard to evaluate each model's graph translation. For both AtomG2G and HierG2G, we've excluded the generated molecules that are too large with more atoms than the maximum number of atoms in our dataset for computing metrics other than validity. ↑ and ↓ denote higher and lower values are better, respectively. The best performance is highlighted in bold.

Model	Type	Validity(↑)	FCD(↓)	NLL(↓)	LogP W_1(↓)	QED MAD(↓)	SAscore MAD(↓)
Reference[1]	-	-	4.807 / 0.279	360.862	2.007	0.153	0.595
AtomG2G	Latent	**100.0**	5.480	347.280	0.222	0.135	0.742
HierG2G	Latent	**100.0**	4.413	344.208	**0.053**	0.135	0.657
DBM	Bridge	89.5	1.038	285.576	0.161	0.145	0.611
DDSBM	Schrödinger Bridge	95.3	**0.896**	**162.228**	0.134	**0.118**	**0.398**

[1] NLL, W_1, and MADs were calculated using random pairs from the test set. Two FCD values are provided: the first compares the initial molecules in the test set with the terminal molecules in the training set, and the second compares the terminal molecules in both sets.

distribution, we report the negative log-likelihood (NLL), which measures the cost of transforming one graph into another based on the reference process, aligning with the graph edit distance (GED) (see Appendix B.6).

For molecular properties, we measure the Wasserstein-1 distance (W_1) between the property distributions of the target and generated molecules representing the properties to be modified, which corresponds to the marginal distribution. In terms of the joint distribution, we calculate the mean absolute difference (MAD) of other key properties between the initial and generated molecules that should be retained, where drug-likeness (QED) [37] and synthetic accessibility score (SAscore) [38] are typical examples in optimizing drug candidates.

5.2 Small molecule transformation

First, we validate our proposed methods on the SB problem between two molecule distributions constructed from the standard ZINC250K dataset. We constructed two sets of molecules whose $\log P$ values are largely different. Molecules from the ZINC250K dataset were randomly selected and divided into two sets whose $\log P$ values follow the Gaussian distributions centered at 2 and 4 with variance of 0.5, respectively (For more details, see Appendix D.1). We compared our methods with three baseline models that perform graph-to-graph translation. AtomG2G and HierG2G are latent-based models that encode an input graph into a latent vector and decode it into another graph [31]. Diffusion Bridge Model (DBM) is a bridge model trained with the same reference process as DDSBM, which is equivalent to the first Markov projection in DDSBM. We refer readers to Appendix C.2 for more details about the baselines and implementation of our models. We note that, for these three baseline models, the initial coupling is utilized during the whole training procedure without change, while DDSBM dynamically alters the training data pair by Iterative Markovian Fitting (IMF).

Table 1 shows overall results of our method compared to the three baseline models. From Table 1, we find that DDSBM constantly outperforms the other models in terms of both FCD and NLL. The lower FCD suggests that DDSBM-generated molecules are more similar to those in the target dataset, while the minimal NLL indicates that DDSBM applies minimal structural change on initial graphs. This result demonstrates that DDSBM achieves more *optimal* graph transformation between fixed initial and terminal distributions. When it comes to the model type, bridge models achieved lower FCD and NLL compared to the latent-based models. The bridge models, DBM and DDSBM, dynamically transform a graph based on the reference dynamics, favoring the retention of the given structure, whereas whole-graph reconstruction using a latent vector does not. This leads to lower NLL values for the bridge models, which is ensured by the Definition 2.2. Additionally, the lower FCD values of the bridge models highlight that constructing a dynamic bridge within the graph domain enhances the performance of distribution learning for the target distribution. Furthermore, we attribute the best performance of DDSBM to the gradual updating of training pairs, which become more similar than the random initial pairs, simplifying the graph transportation process.

Next, we analyze molecular properties of the source and generated molecules. DDSBM resulted in the lowest MAD in QED and SAscore, meaning deviation of molecular properties other than $\log P$ is the smallest for DDSBM. It is noteworthy that DDSBM achieves minimal changes in various molecular properties despite being trained solely to optimize graph transformations with minimal structural alterations, without explicit knowledge on molecular properties. Meanwhile, the latent-based models exhibited larger MAD values in QED and SAscore, indicating they are vulnerable to losing other properties of the initial molecules. This can be inferred from the result of HierG2G; although HierG2G achieves the lowest W_1 value in $\log P$ so that it modulates $\log P$ the best to the desired degree, its much larger NLL value suggests that it might generate a graph with less consideration of the initial graph constraint, as illustrated in Figure 3. For a better understanding of overall results, we provide the corresponding distributions of properties of the models in Figure 5.

Despite the promising results, we observe that all models except DDSBM have a high reliance on a pre-defined initial coupling. Thus, we conducted additional experiments using pseudo-optimal initial coupling based on Tanimoto similarity, which is detailed in Appendix D.3. Interestingly, we found that introducing the pseudo-optimal initial coupling not only accelerated the training of DDSBM in practice but also allowed it to achieve performance on par with the previous results.

5.3 Polymer graph transformation

The Polymer dataset [35] consists of 91,000 monomer molecules with their optical excitation energies (GAPs) obtained by time-dependent density functional theory calculations. We reconstructed the Polymer dataset for a transport problem between two sets of molecules with distinct GAPs, corresponding to green and blue optical colors, respectively. The reconstructed dataset contains 7,603 molecular pairs. This task is considered as a more challenging application because the relationship between the graph structure and the target GAP property is highly non-linear, making it hard to predict the effect of specific structural changes on the GAP. In this context, we apply our DDSBM model to find the optimal transformation between the two sets of molecules.

The performance of DDSBM is compared to DBM, which serves as a baseline. The GAPs of the molecules generated by the models were obtained using the pre-trained MolCLR model [39], which has a mean absolute error (MAE) of 0.027 eV for the GAP prediction. Table 2 shows the overall results of our method on the Polymer dataset. Apparently, DDSBM outperforms DBM on all evaluated metrics, which is consistent with the results from the experiments on ZINC250K. In terms of validity, DBM shows significantly lower scores, which contrasts with the results observed on the ZINC250K dataset. This can be attributed to the characteristics of the molecules in the Polymer dataset, which have relatively large sizes and multiple ring structures. In this context, minimal transformations are advantageous for achieving high validity, and DBM may have struggled to learn these changes from the randomly paired data. Examples of the generated molecular graphs are visualized in Figure 4.

Table 2: **Distribution shift performance on Polymer.** Reference refers to metrics from the initial coupling, used as a standard to evaluate each model's graph translation. ↑ and ↓ denote higher and lower values are better, respectively. The best performance is highlighted in bold.

Model	Type	Validity(↑)	FCD(↓)	NLL(↓)	GAP W_1(↓)
Reference[1]	-	-	1.469 / 0.384	749.800	0.312
DBM	Bridge	35.7	2.803	585.043	0.258
DDSBM	Schrödinger Bridge	**96.6**	**1.101**	**215.590**	**0.136**

[1] NLL and W_1 were calculated using random pairs from the test set. Two FCD values are provided: the first compares the initial molecules in the test set with the terminal molecules in the training set, and the second compares the terminal molecules in both sets.

6 Discussion

In this paper, we propose Discrete Diffusion Schrödinger Bridge Matching (DDSBM), a novel framework utilizing continuous-time Markov chains to solve the SB problem in a high-dimensional discrete state space. To this end, we extend Iterative Markovian Fitting (IMF), proving its convergence to SB. We successfully apply our framework to graph transformation, specifically for molecular optimization. Experimental results demonstrate that DDSBM effectively transforms molecules to achieve the desired property with minimal graph transformation, while retaining other features.

However, our method has several limitations. The IMF requires iterative sampling from the learned Markov process, which can be more computationally intensive than simple bridge matching. Additionally, the graph permutation matching may introduce challenges for wider applicability in general graph tasks.

Acknowledgments and Disclosure of Funding

This work was supported by the Korea Environmental Industry and Technology Institute (Grant No. RS202300219144), the Technology Innovation Program funded by the Ministry of Trade, Industry & Energy, MOTIE, Korea (Grant No. 20016007), and Basic Science Research Programs through the National Research Foundation of Korea funded by the Ministry of Science and ICT (Grant No. RS-2023-00257479 and Grant No. 2018R1A5A1025208).

References

[1] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[2] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[4] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[5] Seonghwan Kim, Jeheon Woo, and Woo Youn Kim. Diffusion-based generative ai for exploring transition states from 2d molecular graphs. *Nature Communications*, 15(1):341, 2024.

[6] Xingchao Liu and Lemeng Wu. Learning diffusion bridges on constrained domains. In *international conference on learning representations (ICLR)*, 2023.

[7] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^{2}sb: Image-to-image schrödinger bridge. *arXiv preprint arxiv:2302.05872*, 2023.

[8] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. *arXiv preprint arXiv:2309.16948*, 2023.

[9] Matteo Pariset, Ya-Ping Hsieh, Charlotte Bunne, Andreas Krause, and Valentin De Bortoli. Unbalanced diffusion schr\" odinger bridge. *arXiv preprint arXiv:2306.09099*, 2023.

[10] Beomsu Kim, Gihyun Kwon, Kwanyoung Kim, and Jong Chul Ye. Unpaired image-to-image translation via neural schr\" odinger bridge. *arXiv preprint arXiv:2305.15086*, 2023.

[11] Xuanzhao Dong, Vamsi Krishna Vasa, Wenhui Zhu, Peijie Qiu, Xiwen Chen, Yi Su, Yujian Xiong, Zhangsihao Yang, Yanxi Chen, and Yalin Wang. Cunsb-rfie: Context-aware unpaired neural schr"{o} dinger bridge in retinal fundus image enhancement. *arXiv preprint arXiv:2409.10966*, 2024.

[12] Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pages 269–310, 1932.

[13] Christian Léonard. A survey of the schr\" odinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

[14] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[15] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

[16] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[17] Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos Theodorou. Deep generalized schrödinger bridge. *Advances in Neural Information Processing Systems*, 35:9374–9388, 2022.

[18] Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion schrödinger bridges. *arXiv preprint arxiv:2302.11419*, 2023.

[19] Stefano Peluchetti. Diffusion bridge mixture transports, schrödinger bridge problems and generative modeling. *Journal of Machine Learning Research*, 24(374):1–51, 2023.

[20] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] Danyeong Lee, Dohoon Lee, Dongmin Bang, and Sun Kim. Disco: Diffusion schrödinger bridge for molecular conformer optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(12):13365–13373, March 2024. ISSN 2159-5399. doi: 10.1609/aaai.v38i12.29238. URL `http://dx.doi.org/10.1609/aaai.v38i12.29238`.

[22] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[23] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

[24] Alberto Sanfeliu and King-Sun Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (3):353–362, 1983.

[25] Christian Léonard, Sylvie Rœlly, and Jean-Claude Zambrini. Reciprocal processes. a measure-theoretical point of view. 2014.

[26] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[27] Minsu Cho, Jian Sun, Olivier Duchenne, and Jean Ponce. Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2083–2090, 2014.

[28] Ilia Igashov, Arne Schneuing, Marwin Segler, Michael Bronstein, and Bruno Correia. Retrobridge: Modeling retrosynthesis with markov bridges. *arXiv preprint arXiv:2308.16212*, 2023.

[29] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1720–1733, 2023.

[30] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018.

[31] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International conference on machine learning*, pages 4839–4848. PMLR, 2020.

[32] Ziqi Chen, Martin Renqiang Min, Srinivasan Parthasarathy, and Xia Ning. A deep generative model for molecule optimization via one fragment modification. *Nature machine intelligence*, 3 (12):1040–1049, 2021.

[33] Hyeongwoo Kim, Seokhyun Moon, Wonho Zhung, Jaechang Lim, and Woo Youn Kim. Deepbioisostere: Discovering bioisosteres with deep learning for a fine control of multiple molecular properties. *arXiv preprint arXiv:2403.02706*, 2024.

[34] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International conference on machine learning*, pages 1945–1954. PMLR, 2017.

[35] Peter C. St. John, Caleb Phillips, Travis W. Kemper, A. Nolan Wilson, Yanfei Guan, Michael F. Crowley, Mark R. Nimlos, and Ross E. Larsen. Message-passing neural networks for high-throughput polymer screening. *The Journal of Chemical Physics*, 150(23), June 2019. ISSN 1089-7690.

[36] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Gunter Klambauer. Fréchet chemnet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58(9):1736–1741, 2018.

[37] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

[38] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11, 2009.

[39] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3): 279–287, 2022.

[40] Jean-René Chazottes, Cristian Giardina, and Frank Redig. Relative entropy and waiting times for continuous-time markov processes. 2006.

[41] Claude Kipnis and Claudio Landim. *Scaling limits of interacting particle systems*, volume 320. Springer Science & Business Media, 2013.

[42] Benton Jamison. Reciprocal processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 30(1):65–86, 1974.

[43] Stefano Peluchetti. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2023.

[44] Patrick Billingsley. *Convergence of probability measures*. John Wiley & Sons, 2013.

[45] Stewart N Ethier and Thomas G Kurtz. *Markov processes: characterization and convergence*. John Wiley & Sons, 2009.

[46] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565, 1976.

[47] Kurt M Anstreicher. Recent advances in the solution of quadratic assignment problems. *Mathematical Programming*, 97:27–42, 2003.

[48] Paul C Gilmore. Optimal and suboptimal algorithms for the quadratic assignment problem. *Journal of the society for industrial and applied mathematics*, 10(2):305–313, 1962.

[49] Eliane Maria Loiola, Nair Maria Maia De Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690, 2007.

[50] Yong Xia. Gilmore-lawler bound of quadratic assignment problem. *Frontiers of Mathematics in China*, 3:109–118, 2008.

[51] Marius Leordeanu and Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1482–1489. IEEE, 2005.

[52] Sébastien Bougleux, Luc Brun, Vincenzo Carletti, Pasquale Foggia, Benoit Gaüzere, and Mario Vento. A quadratic assignment formulation of the graph edit distance. *arXiv preprint arXiv:1512.07494*, 2015.

[53] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

[54] Peter Willett, John M Barnard, and Geoffrey M Downs. Chemical similarity searching. *Journal of chemical information and computer sciences*, 38(6):983–996, 1998.

A Propositions and proof

A.1 Notations

In this section, we introduce the notations that will be used throughout the proofs of the propositions. $\Omega = D([0,\tau],\mathcal{X})$ denotes the space of all left-limited and right-continuous (càdlàg) paths over a finite space $\mathcal{X}$. We assume the state space $\mathcal{X}$ has connected finite graph structure (fully-connected graph), which imply it becomes metric space with graph distance metric $d_{\mathcal{X}}(\cdot,\cdot)$. Accordingly, the sample space Ω is equipped with Skorokhod topology with Skorokhod metric $d_{\Omega}(\cdot,\cdot)$, and associated Borel σ-algebra. $X = (X_t)_{t\in[0,\tau]}$ denotes the canonical process given by:

$$X_t(\omega) = \omega_t, \quad t \in [0,\tau], \quad \omega = (\omega_s)_{s\in[0,\tau]}.$$

The reference measure $\mathbb{Q}$ is an irreducible Markov measure on Ω with its associated canonical filtration. Assume that the transition probability of $\mathbb{Q}$, denoted by $P_{s:t}(x,y)$, from $(s,x) \in [0,\tau]\times\mathcal{X}$ to $(t,y) \in [0,\tau]\times\mathcal{X}$, is continuous and differentiable over time. The measure is generated by the transition rate function $A_s(x,y)$, which gives the rate of transition from $x \in \mathcal{X}$ to $y \in \mathcal{X}$ at time $s \in (0,\tau)$, and satisfies the Kolmogorov forward equation:

$$\frac{\partial P_{s:t}(x,y)}{\partial t} = \sum_{z\in\mathcal{X}} P_{s:t}(x,z)A_t(z,y), \tag{9}$$

$$A_s(x,y) = \left[\frac{\partial P_{s:t}(x,y)}{\partial t}\right]_{t=s}.$$

We also assume that $\mathbb{Q}$ can construct a bridge $\mathbb{Q}(\cdot|X_0 = x, X_\tau = y)$ for all $x,y \in \mathcal{X}$. For any Markov measure M, the corresponding generator is denoted as $A^{(M)}$.

A.2 Theorem A.1

Theorem A.1. *(Uniqueness of the Schrödinger Bridge Solution)*
If the reference process $\mathbb{Q}$ is Markov, then under mild conditions the Schrödinger Bridge solution $\mathbb{P}^{SB}$ exists and is unique. Furthermore, the solution is mixture with static Schrödinger Bridge solution represented as $\mathbb{P}^{SB} = \mathbb{P}^{SB}_{0,\tau}\mathbb{Q}_{\cdot|0,\tau} \in \mathcal{R}(\mathbb{Q})$, and also it is in $\mathcal{M}$. Conversely, a process $\mathbb{P} = \mathbb{P}_{0,\tau}\mathbb{Q}_{\cdot|0,\tau}$ is Markov if and only if $\mathbb{P} = \mathbb{P}^{SB}$.

Proof. This is direct consequence of Theorem 2.12 of [13]. □

A.3 Markov projection

Proposition A.2. *(solution of Markov projection)*
Let $M^* = \Pi_{\mathcal{M}}(\Lambda)$, $\Lambda \in \mathcal{R}(\mathbb{Q})$ and the generator of $\mathbb{Q}$ be $A_t(x,y)$ with transition probability $P_{s:t}(x,y)$. Under mild assumptions, the generator of the Markov measure M^* becomes

$$A_t^{(M^*)}(X_t,y) = \mathbb{E}_{\Lambda_{\tau|t}}\left[A_t(X_t,y;X_\tau)\middle|X_t\right]$$

$$A_t(x,y;z) = A_t(x,y)\frac{P_{s:\tau}(y,z)}{P_{s:\tau}(x,z)} - \delta_{xy}\sum_u A_t(y,u)\frac{P_{t:\tau}(u,z)}{P_{t:\tau}(x,z)}$$

, where $z \in \mathcal{X}$

The reverse KL-divergence is

$$D_{KL}(\Lambda\|M^*) = \int_0^\tau \mathbb{E}_{\Lambda_{0,t}}\left[(A_t^{(\Lambda_{|0})} - A_t^{(M^*)})(X_t,X_t) + \sum_{y\neq X_t} A_t^{(\Lambda_{|0})}\log\frac{A_t^{(\Lambda_{|0})}}{A_t^{(M^*)}}(X_t,y)\right]dt,$$

where the $A^{\Lambda_{|0}}$ is the generator for the conditioned measure $\Lambda_{|0}$ which is defined as

$$A_t^{\Lambda_{|0}}(X_t,y) = \mathbb{E}_{\Lambda_{\tau|0,t}}\left[A_t(X_t,y;X_\tau)\middle|X_t,X_0\right].$$

Moreover, for any $t \in [0,\tau]$, $\Lambda_t = M^*_t$.

A.3.1 KL-divergence of Markov measure

Consider two Markov path measure $\tilde{M} \ll M$. Based on the Girsanov's formula, we can express the Radon-Nikodym derivative as follow:

$$\frac{dM}{d\tilde{M}}(\omega) = \frac{dM_0}{d\tilde{M}_0}(\omega_0) \exp\left(\int_0^\tau \log \frac{A_t^M}{A_t^{\tilde{M}}}(\omega_{t-}, \omega_t) dN_t(\omega) + \int_0^\tau (A_t^M - A_t^{\tilde{M}})(\omega_t, \omega_t) dt \right),$$

where $N_t(\omega)$ is the number of jumps of the path ω up to time t and ω_{t-} is the left limit of the path at time t (see [40] or Appendix 1 of [41]). Note that, due to the compactness of time interval, the number of jumps of each path in Ω is at most finite, and thus $N_t(\omega)$ is bounded. Also, the escape rate of the state x associated to M is $-A_t^M(x,x)$. Thus, we can construct a natural martingale (Lemma 5.1 of [41])

$$N_t + \int_0^t A_s(\omega_s, \omega_s) ds,$$

which is zero-mean process.

For any continuous and bounded function $\phi : \mathcal{X} \to \mathbb{R}$, we can change the integrator dN_t as follow:

$$\mathbb{E}_M \left[\int_0^t \phi(\omega_s) dN_s \right] = \mathbb{E}_M \left[\int_0^t -\phi(\omega_s) A_s^M(\omega_s, \omega_s) ds \right] = \int_0^t \mathbb{E}_{x \sim M_s} \left[-\phi(x) A_s^M(x,x) \right] ds.$$

The KL-divergence is expectation of logarithm of Radon-Nikodym derivative, which leads to:

$$\begin{aligned} D_{\mathrm{KL}}(M \| \tilde{M}) &= D_{\mathrm{KL}}(M_0 \| \tilde{M}_0) + \mathbb{E}_M \left[\int_0^\tau \log \frac{A_t^M}{A_t^{\tilde{M}}}(\omega_{t-}, \omega_t) dN_t(\omega) + \int_0^\tau (A_t^M - A_t^{\tilde{M}})(\omega_t, \omega_t) dt \right], \\ &= D_{\mathrm{KL}}(M_0 \| \tilde{M}_0) + \int_0^\tau \mathbb{E}_{x \sim M_s} \left[-A_s^M(x,x) \sum_{y \neq x} p_s(x,y) \log \frac{A_s^M}{A_s^{\tilde{M}}}(x,y) \right] ds \\ &\quad + \int_0^\tau \mathbb{E}_{x \sim M_s} \left[(A_t^M - \tilde{A}_t^{\tilde{M}})(x,x) \right] ds, \end{aligned}$$

where $p_s(x,y)$ is the probability of jump from x to y given that a jump occurs, which is $\frac{A_s^M(x,y)}{-A_s^M(x,x)}$ By applying this, we obtain KL-divergence represented solely in terms of transition rate A^M and $A^{\tilde{M}}$:

$$D_{\mathrm{KL}}(M \| \tilde{M}) = D_{\mathrm{KL}}(M_0 \| \tilde{M}_0) + \int_0^\tau \mathbb{E}_{x \sim M_s} \left[\sum_{y \neq x} A_s^M(x,y) \log \frac{A_s^M}{A_s^{\tilde{M}}}(x,y) + (A_t^M - A_t^{\tilde{M}})(x,x) \right] ds. \tag{10}$$

A.3.2 KL-divergence of reciprocal measure to Markov measure

Lemma A.3. *If a reciprocal measure $\Lambda \in \mathcal{R}(\mathbb{Q})$ is conditioned on X_0 being a.s. constant, then the corresponding measure $\Lambda_{\cdot|0}$ is Markov. For any $M \in \mathcal{M}$ such that $M_0 = \Lambda_0$ and $\Lambda \ll M$, the KL-divergence $D_{KL}(\Lambda \| M)$ disintegrates as follow:*

$$D_{KL}(\Lambda \| M) = \mathbb{E}_{\Lambda_0} \left[D_{KL}(\Lambda_{\cdot|0} \| M_{\cdot|0}) \right].$$

Proof. According to proposition 2.5 of [25] and lemma 1.4 of [42], conditioning Λ on X_0 not only preserves its reciprocal property, but also transforms it into a Markov process. Due to the absolute continuity together with $|\mathcal{X}| < \infty$, the KL-divergence is finite. Then, the KL-divergence can be reformulated as follows:

$$\begin{aligned} D_{\mathrm{KL}}(\Lambda \| M) &= \mathbb{E}_\Lambda \left[\frac{d\Lambda}{dM} \right], \\ &= \mathbb{E}_\Lambda \left[\frac{d\Lambda_{\cdot|0}}{dM_{\cdot|0}} \right], \\ &= \mathbb{E}_{\Lambda_0} \mathbb{E}_{\Lambda_{\cdot|0}} \left[\frac{d\Lambda_{\cdot|0}}{dM_{\cdot|0}} \right], \\ &= \mathbb{E}_{\Lambda_0} \left[D_{\mathrm{KL}}(\Lambda_{\cdot|0} \| M_{\cdot|0}) \right]. \end{aligned}$$

□

According to Lemma A.3, while a reciprocal measure $\Lambda \in \mathcal{R}(\mathbb{Q})$ is in general non-Markov, the conditional measure $\Lambda_{\cdot|0}$ is Markov. Note that we can compute the KL-divergence between two Markov measure based on Equation (10).

A.3.3 Generator of conditioned process

To compute $D_{\text{KL}}(\Lambda\|M)$ based on Lemma A.3, Equation (10), we need the generator of the conditioned process $\Lambda_{\cdot|0}$. Before deriving the generator of $\Lambda_{\cdot|0}$, we first consider the pinned process of $\mathbb{Q}$ conditioned on $X_\tau = z$ in prior.

Lemma A.4. *Let $(X_t)_{0\leq t\leq\tau}$ be a Markov process under the reference measure $\mathbb{Q}$ with transition probability $P_{s:t}(\cdot,\cdot), (s \leq t)$, and generator $A_s(\cdot,\cdot)$. Consider the process conditioned on $X_\tau = z$ with corresponding measure denoted by $\mathbb{Q}^{(z)}$. Then, the conditioned process is also Markov, and its generator is given by:*

$$A_s(x, y; z) = A_s(x, y)\frac{P_{s:\tau}(y, z)}{P_{s:\tau}(x, z)} - \delta_{xy}\left[\sum_u A_s(x, u)\frac{P_{s:\tau}(u, z)}{P_{s:\tau}(x, z)}\right].$$

Proof. The conditional probability of X_t given the natural filtration $\mathcal{F}_u$ and X_s with $u \leq s \leq t \leq \tau$ under the measure $\mathbb{Q}^{(z)}$ is as follows:

$$\begin{aligned}\mathbb{Q}^{(z)}(X_t = y|X_s = x, \mathcal{F}_u) &= \mathbb{Q}(X_t = y|X_s = x, X_\tau = z, \mathcal{F}_u),\\ &= \mathbb{Q}(X_t = y|X_s = x, X_\tau = z), \qquad (\because \mathbb{Q} \in \mathcal{M})\\ &= \mathbb{Q}^{(z)}(X_t = y|X_s = x),\end{aligned}$$

which confirms $\mathbb{Q}^{(z)}$ is Markov.

Next, the transition probability of $\mathbb{Q}^{(z)}$, denoted by $P_{s:t}(x, y; z)$, is derived as:

$$\begin{aligned}P_{s:t}(x, y; z) &= \mathbb{Q}^{(z)}(X_t = y|X_s = x),\\ &= \mathbb{Q}(X_t = y|X_s = x)\frac{\mathbb{Q}(X_\tau = z|X_t = y)}{\mathbb{Q}(X_\tau = z|X_s = x)},\\ &= P_{s:t}(x, y)\frac{P_{t:\tau}(y, z)}{P_{s:\tau}(x, z)}.\end{aligned}$$

Note that, we assumed the measure $\mathbb{Q}$ construct bridge everywhere, the probability ratio has finite value.

Finally, the corresponding generator $A_s(x, y; z)$ is obtained using the Kolmogorov forward equation:

$$\begin{aligned}A_s(x, y; z) &= \left.\frac{\partial}{\partial t}P_{s:t}(x, y; z)\right|_{t=s},\\ &= A_s(x, y)\frac{P_{s:\tau}(y, z)}{P_{s:\tau}(x, z)} + \delta_{xy}\left[\frac{\partial_s P_{s:\tau}(y, z)}{P_{s:\tau}(x, z)}\right],\\ &= A_s(x, y)\frac{P_{s:\tau}(y, z)}{P_{s:\tau}(x, z)} - \delta_{xy}\left[\sum_u A_s(x, u)\frac{P_{s:\tau}(u, z)}{P_{s:\tau}(x, z)}\right].\end{aligned}$$

□

We now consider the transition probability and generator of conditioned measure $\Lambda_{\cdot|0}$ of $\Lambda \in \mathcal{R}(\mathbb{Q})$.

Lemma A.5. *For a reciprocal measure $\Lambda \in \mathcal{R}(\mathbb{Q})$, the conditioned process with $X_0 = x_0$ is denoted by $\Lambda_{\cdot|0=x_0}$. The generator of $\Lambda_{\cdot|0=x_0}$ is given by the conditional expectation:*

$$A_s^{\Lambda|0=x_0}(x, y) = \mathbb{E}_{z\sim\Lambda_{\tau|0,s}}\left[A_s(x, y; z)|X_0 = x_0, X_s = x\right],$$

where $A_s(x, y; z)$ is the generator of the conditioned process of $\mathbb{Q}$ with $X_\tau = z$.

Proof. We denote the transition probability conditioned on $X_0 = x_0$ by $P_{s:t}^{\Lambda_{\cdot|0=x_0}}$, which is computed as follows:

$$P_{s:t}^{\Lambda_{\cdot|0=x_0}}(x,y) = \int_{\mathcal{X}} \Lambda_{\cdot|0=x_0}(X_\tau = z|X_s = x)\Lambda_{\cdot|0=x_0}(X_t = y|X_s = x, X_\tau = z)dz.$$

The first term in the integrand is

$$\begin{aligned}
\Lambda_{\cdot|0=x_0}(X_\tau = z|X_s = x) &= \Lambda(X_\tau = z|X_0 = x_0, X_s = x), \\
&= \frac{\Lambda(X_\tau = z|X_0 = x_0)\Lambda(X_s = x|X_0 = x_0, X_\tau = z)}{\Lambda(X_s = x|X_0 = x_0)}, \\
&= \frac{\nu_\tau(z;x_0)}{\nu_s(x;x_0)} P_{0:s}(x_0, x; z), \\
&= \frac{\nu_\tau(z;x_0)}{\nu_s(x;x_0)} P_{0:s}(x_0, x) \frac{P_{s:\tau}(x,z)}{P_{0:\tau}(x_0,z)}, \\
&= \frac{\nu_\tau(z;x_0)}{\nu_s(x;x_0)} \frac{\mu_s(x;x_0)}{\mu_\tau(z;x_0)} P_{s:\tau}(x,z),
\end{aligned}$$

where $\nu_s(\cdot;x_0), \mu_s(\cdot;x_0)$ are the probability mass functions of $\Lambda_{s|0}, \mathbb{Q}_{s|0}$, respectively, with $X_0 = x_0$. Given the initial-terminal condition, Λ is equivalent to the reference $\mathbb{Q}$ based on the definition of reciprocal class Definition 2.1. Similarly, the second term is same as $\mathbb{Q}(X_t = y|X_0 = x_0, X_s = x, X_\tau = z)$, which can be expressed as $P_{s:t}(x,y;z)$ due to the Markov property of $\mathbb{Q}$. Therefore, the transition probability is:

$$P_{s:t}^{\Lambda_{\cdot|0=x_0}}(x,y) = \frac{\mu_s(x;x_0)}{\nu_s(x;x_0)} P_{s:t}(x,y) \int_{\mathcal{X}} P_{t:\tau}(y,z) \frac{\nu_\tau(z;x_0)}{\mu_\tau(z;x_0)} dz.$$

Accordingly, the generator is derived as:

$$\begin{aligned}
A_s^{\Lambda_{\cdot|0=x_0}}(x,y) &= \partial_t P_{s:t}^{\Lambda_{\cdot|0=x_0}}(x,y)\Big|_{t=s}, \\
&= \frac{\mu_s(x;x_0)}{\nu_s(x;x_0)} \partial_t P_{s:t}(x,y)\Big|_{t=s} \int_{\mathcal{X}} P_{t:\tau}(y,z) \frac{\nu_\tau(z;x_0)}{\mu_\tau(z;x_0)} dz \\
&\quad + \frac{\mu_s(x;x_0)}{\nu_s(x;x_0)} P_{s:t}(x,y) \int_{\mathcal{X}} \partial_t P_{t:\tau}(y,z)\Big|_{t=s} \frac{\nu_\tau(z;x_0)}{\mu_\tau(z;x_0)} dz, \\
&= \frac{\mu_s(x;x_0)}{\nu_s(x;x_0)} A_s(x,y) \int_{\mathcal{X}} P_{s:\tau}(y,z) \frac{\nu_\tau(z;x_0)}{\mu_\tau(z;x_0)} dz \\
&\quad - \delta_{xy} \frac{\mu_s(x;x_0)}{\nu_s(x;x_0)} \int_{\mathcal{X}} \sum_u \left[A_s(y,u) P_{s:\tau}(u,z)\right] \frac{\nu_\tau(z;x_0)}{\mu_\tau(z;x_0)} dz, \\
&= \int_{\mathcal{X}} \frac{\mu_s(x;x_0)}{\nu_s(x;x_0)} \left[A_s(x,y) P_{s:\tau}(y,z) - \delta_{xy} \sum_u A_s(y,u) P_{s:\tau}(u,z)\right] \frac{\nu_\tau(z;x_0)}{\mu_\tau(z;x_0)} dz, \\
&= \int_{\mathcal{X}} \frac{\nu_\tau(z;x_0)}{\mu_\tau(z;x_0)} \frac{\mu_s(x;x_0)}{\nu_s(x;x_0)} P_{s:\tau}(x,z) A_s(x,y;z) dz, \\
&= \int_{\mathcal{X}} \Lambda_{|0}(X_\tau = z|X_s = x) A_s(x,y;z) dz, \\
&= \mathbb{E}_{z\sim\Lambda_{\tau|0,s}}[A_s(x,y;z)|X_0 = x_0, X_s = x].
\end{aligned}$$

In conclusion, the generator of $\Lambda_{\cdot|0}$ is given as the conditional expectation,

$$A_s^{\Lambda_{\cdot|0}}(x,y) = \mathbb{E}_{\Lambda_{\tau|0,s}}[A_s(x,y;X_\tau)|X_0, X_s = x].$$

□

A.3.4 Marginal distribution of mixture of Markov chains

A reciprocal process $\Lambda \in \mathcal{R}(\mathbb{Q})$ can be represented as a mixture of Markov chains as:

$$\Lambda(\cdot) = \sum_{x,y} \mathbb{Q}(\cdot|X_0 = x, X_\tau = y)\Lambda_{0,\tau}(x,y).$$

We here propose the mixture representation of Markov chain similar to theorem 2 of [43] which describing mixture of diffusion process over Euclidean space.

Proposition A.6. *Consider a family of Markov measures on Ω indexed by $u \in \mathcal{I}$*

$$\begin{aligned}\partial_t P^u_{s:t}(x,y) &= \sum_z P^u_{s:t}(x,z)A^u_t(z,y),\\ A^u_s(x,y) &= \partial_t P^u_{s,t}(x,y)\bigg|_{t=s},\end{aligned}$$

where P^u and A^u is the transition probability and generator for each measure associated to the process $X^u = (X^u_t)_{t\in[0,\tau]}$. We here assume that A^u_s is finite for every u. Let the corresponding Markov measure be M^u, and μ^u_t be the density of M^u_t, the marginal distribution of X^u_t. Let a mixture of M^u with the index distribution $\mathcal{U}$ over $\mathcal{I}$ be Λ,

$$\Lambda(\cdot) = \int_{\mathcal{I}} M^u(\cdot)\mathcal{U}(du).$$

We denote mixture marginal density μ_t and mixture initial distribution Λ_0 as:

$$\begin{aligned}\mu_t(x) &= \int_{\mathcal{I}} \mu^u_t(x)\mathcal{U}(du),\\ \Lambda_0(\cdot) &= \int_{\mathcal{I}} M^u_0(\cdot)\mathcal{U}(du).\end{aligned}$$

Let $X = (X_t)_{t\in[0,\tau]}$ be another Markov chain generated by:

$$\begin{aligned}&\partial_t P_{s:t}(x,y) = \sum_z P_{s:t}(x,z)A_t(z,y),\\ &A_t(x,y) = \frac{1}{\mu_t(x)}\int_{\mathcal{I}} A^u_t(x,y)\mu^u_t(x)\mathcal{U}(du),\\ &X_0 \sim \Lambda_0.\end{aligned}$$

Then, the marginal density of X_t is μ_t. It is assumed that exchange of ∂_t and $\int_{\mathcal{I}}\mathcal{U}(du)$ is justified.

Proof. We start from verifying $A_t(x,y)$ admits conditions of transition rate function. For $x \neq y$ it is trivial that $A_t(x,y)$ is finite and non-negative. Also, $\sum_{y\in\mathcal{X}} A_t(x,y)$ becomes zeros for all x and t as:

$$\begin{aligned}\sum_{y\in\mathcal{X}} A_t(x,y) &= \sum_{y\in\mathcal{X}} \frac{1}{\mu_t(y)}\int_{\mathcal{I}} A^u_t(x,y)\mu^u_t(x)\mathcal{U}(du),\\ &= \frac{1}{\mu_t(y)}\int_{\mathcal{I}} \sum_{y\in\mathcal{X}} A^u_t(x,y)\mu^u_t(x)\mathcal{U}(du),\\ &= 0.\end{aligned}$$

Thus, mixture of generators $A_t(x,y)$ holds the condition for a generator of Markov measures.

Next, for $t \in (0, \tau)$,

$$\begin{aligned}
\partial_t \mu_t(x) &= \partial_t \int_{\mathcal{I}} \mu_t^u(x) \mathcal{U}(du), \\
&= \int_{\mathcal{I}} \partial_t \mu_t^u(x) \mathcal{U}(du), && \text{(assumption)} \\
&= \int_{\mathcal{I}} \sum_{y \in \mathcal{X}} A_t^u(y, x) \mu_t^u(y) \mathcal{U}(du), && (\because \text{Kolmogorov equation}) \\
&= \sum_{y \in \mathcal{X}} \left(\int_{\mathcal{I}} A_t^u(y, x) \mu_t^u(y) \mathcal{U}(du) \right), \\
&= \sum_{y \in \mathcal{X}} A_t(y, x) \mu_t(y).
\end{aligned}$$

The equality of the left-hand side and the final line corresponds to the Kolmogorov equation for the process X generated by A_t. This shows that μ_t is the marginal distribution of X_t. □

A.3.5 Minimizer of the KL-divergence

The Markov projection of a reciprocal process Λ, denoted $\Pi_{\mathcal{M}}(\Lambda)$, is a Markov process M^* which minimizes the reverse KL-divergence $D_{\text{KL}}(\Lambda, M)$. We here characterize M^* by specifying its generator. While the generator $A_t^{\Lambda_{\cdot|0}}$ of $\Lambda_{\cdot|0}$ represented as a conditional expectation given X_0, X_t as noted in Lemma A.5, that of the M^* is supposed to be represented as the conditional expectation without X_0 condition. The following lemma says that $\mathbb{E}_{\Lambda_{0|t}} \left[A_t^{\Lambda_{\cdot|0}}(X_t, y) \right]$ is the generator.

Lemma A.7. *Let the Markov projection of a reciprocal process $\Lambda \in \mathcal{R}(\mathbb{Q})$ be $M^* = \Pi_{\mathcal{M}}(\Lambda)$. Then the generator of M^* is $A_t^{M^*}(X_t, y) = \mathbb{E}_{\Lambda_{\tau|t}} \left[A_t(X_t, y; X_\tau) \middle| X_t \right]$.*

Proof. Because $\mathbb{Q}$ is assumed to be able to construct bridge everywhere, $A_t(x, y; z) = 0 \iff A_t(x, y) = 0 \quad \forall x \neq y, z \in \mathcal{X}, t \in [0, \tau)$.

We claim that the Markov measure M generated by $A_t^M(x, y) = \mathbb{E}_{\Lambda_{\tau|t}} \left[A_t(X_t, y; X_\tau) \middle| X_t = x \right]$ with $M_0 = \Lambda_0$ is a.s. M^*. We can re-formulate the generator as:

$$\begin{aligned}
A_t^M(X_t, y) &= \mathbb{E}_{\Lambda_{\tau|t}} \left[A_t(X_t, y; X_\tau) \middle| X_t \right], \\
&= \sum_{x_\tau} \Lambda(X_\tau = x_\tau | X_t) A_t(X_t, y; x_\tau), \\
&= \sum_{x_\tau} \sum_{x_0} \Lambda(X_0 = x_0, X_\tau = x_\tau | X_t) A_t(X_t, y; x_\tau), \\
&= \sum_{x_0} \Lambda(X_0 = x_0 | X_t) \sum_{x_\tau} \Lambda(X_\tau = x_\tau | X_0 = x_0, X_t) A_t(X_t, y; x_\tau), \\
&= \sum_{x_0} \Lambda(X_0 = x_0 | X_t) A_t^{\Lambda_{\cdot|0}}(X_t, y), \\
&= \mathbb{E}_{\Lambda_{0|t}} \left[A_t^{\Lambda_{\cdot|0}}(X_t, y) \middle| X_t \right], && (11)
\end{aligned}$$

which conclude that it becomes conditional expectation of the generator of $\Lambda_{\cdot|0}$. Also, based on the Proposition A.6, the $M_t = \Lambda_t$ for all t.

Note that $A_t^{\Lambda_{\cdot|0}}(x, y) = \mathbb{E}_{\Lambda_{\tau|0,t}} \left[A_t(x, y; X_\tau) \middle| X_0, X_t = x \right]$, which implies $A_t^M(x, y) = 0 \implies A_t^{\Lambda_{\cdot|0}}(x, y) = 0 \quad \forall x \neq y, z \in \mathcal{X}, t \in [0, \tau)$. Thus, $\Lambda \ll M$. Because M^* is the minimizer of KL-divergence $D_{\text{KL}}(\Lambda \| \cdot)$, we can assume $M_0^* = \Lambda_0$. Based on this, we want to show that

$D_{\mathrm{KL}}(\Lambda\|M^*) - D_{\mathrm{KL}}(\Lambda\|M) = 0$. Recall that Equation (10), where the KL-divergence is formulated as

$$\begin{aligned} D_{\mathrm{KL}}(\Lambda\|M) &= \int_0^\tau \mathbb{E}_{\Lambda_{0,t}}\left[\left(A_t^{\Lambda_{\cdot|0}} - A_t^M\right)(X_t, X_t) + \sum_{y\neq X_t} A_t^{\Lambda_{\cdot|0}} \log \frac{A_t^{\Lambda_{\cdot|0}}}{A_t^M}(X_t, y)\right] dt, \\ &= \int_0^\tau \underbrace{\mathbb{E}_{\Lambda_{0,t}}\left[\sum_{y\neq X_t}\left(A_t^{\Lambda_{\cdot|0}} \log \frac{A_t^{\Lambda_{\cdot|0}}}{A_t^M} - A_t^{\Lambda_{\cdot|0}} + A_t^M\right)(X_t, y)\right]}_{f(t,\Lambda,M)} dt. \end{aligned}$$

Then, $D_{\mathrm{KL}}(\Lambda\|M^*) - D_{\mathrm{KL}}(\Lambda\|M) = \int_0^\tau \Delta dt \leq 0$ by definition, where $\Delta := f(t, \Lambda, M^*) - f(t, \Lambda, M)$.

$$\begin{aligned} \Delta &= \mathbb{E}_{\Lambda_{0,t}}\left[\sum_{y\neq X_t}\left(A_t^{M^*} - A_t^M + A_t^{\Lambda_{\cdot|0}} \log \frac{A_t^M}{A_t^{M^*}}\right)(X_t, y)\right], \\ &= \mathbb{E}_{\Lambda_t}\mathbb{E}_{\Lambda_{0|t}}\left[\sum_{y\neq X_t}\left(A_t^{M^*} - A_t^M + A_t^{\Lambda_{\cdot|0}} \log \frac{A_t^M}{A_t^{M^*}}\right)(X_t, y)\right], \\ &= \mathbb{E}_{\Lambda_t}\sum_{y\neq X_t}\mathbb{E}_{\Lambda_{0|t}}\left[\left(A_t^{M^*} - A_t^M + A_t^{\Lambda_{\cdot|0}} \log \frac{A_t^M}{A_t^{M^*}}\right)(X_t, y)\right], \\ &= \mathbb{E}_{\Lambda_t}\sum_{y\neq X_t}\left(A_t^{M^*} - A_t^M + \mathbb{E}_{\Lambda_{0|t}}\left[A_t^{\Lambda_{\cdot|0}}|X_t\right] \log \frac{A_t^M}{A_t^{M^*}}\right)(X_t, y), \qquad (\because M, M^* \in \mathcal{M}) \\ &= \mathbb{E}_{\Lambda_t}\sum_{y\neq X_t}\left(A_t^{M^*} - A_t^M + A_t^M \log \frac{A_t^M}{A_t^{M^*}}\right)(X_t, y), \qquad (\because \mathit{Equation}\ (11)) \\ &= \mathbb{E}_{M_t}\sum_{y\neq X_t}\left(A_t^{M^*} - A_t^M + A_t^M \log \frac{A_t^M}{A_t^{M^*}}\right)(X_t, y). \qquad (\because M_t = \Lambda_t) \end{aligned}$$

We can deduce that $\int_0^\tau \Delta dt = D_{\mathrm{KL}}(M\|M^*) \geq 0$, which conclude that $M = M^*$.

□

A.3.6 Proof of Proposition A.2

Now we prove the proposition Proposition A.2 using above lemmas.

Proof. Proposition A.2

By the Lemma A.3 and Equation (10), the KL-divergence of Λ to any Markov measure $M \in \mathcal{M}$ disintegrates as:

$$D_{\mathrm{KL}}(\Lambda\|M) = \int_0^\tau \mathbb{E}_{\Lambda_{0,t}}\left[\left(A_t^{\Lambda_{\cdot|0}} - A_t^M\right)(X_t, X_t) + \sum_{y\neq X_t} A_t^{\Lambda_{\cdot|0}} \log \frac{A_t^{\Lambda_{\cdot|0}}}{A_t^M}(X_t, y)\right] dt,$$

where the $A_t^{\Lambda_{\cdot|0}}$ is the generator of pinned process of $\Lambda_{\cdot|0}$ stated by Lemma A.5. According to Lemma A.7, the Markov measure M^* generated by $A_t(x, y) = \mathbb{E}_{\Lambda_{\tau|t}}\left[A_t(x, y; X_\tau)|X_t = x\right]$ is the minimizer of $D_{\mathrm{KL}}(\Lambda\|M)$ for $M \in \mathcal{M}$. In last, Proposition A.6 ensure the time marginals of Λ, M^* for all $t \in [0, \tau]$ are equivalent. □

A.4 Time-reversal Markov projection

Proposition A.8. *Let $M^* = \Pi_{\mathcal{M}}(\Lambda), \Lambda \in \mathcal{R}(\mathbb{Q})$ and the forward generator of $\mathbb{Q}$ be $A_t(x, y)$ with the transition probability $P_{s:t}(x, y)$. Let $X = (X_t)_{t\in[0,\tau]}$ be the canonical process and $Y = (Y_t)_{t\in[0,\tau]}$ be the time reversal of X, where $Y_t = X_{\tau-t}$. Under mild assumptions, the time-reversal generator of M^* becomes*

$$\begin{aligned}\tilde{A}_t^{M^*}(Y_t, x) &= \mathbb{E}_{0|t}\left[\tilde{A}_t(Y_t, x; X_0)\middle|Y_t\right]\\ \tilde{A}_s(y, x; z) &= \partial_t \tilde{P}_{s:t}(y, x; z)\bigg|_{t=s}\\ &= A_{\tau-s}(x, y)\frac{P_{0:\tau-s}(z, x)}{P_{0:\tau-s}(z, y)} - \delta_{xy}\sum_u A_{\tau-s}(u, x)\frac{P_{0:\tau-s}(z, u)}{P_{0:\tau-s}(z, x)},\end{aligned}$$

where $z \in \mathcal{X}$, and $\tilde{A}_s(\cdot, \cdot; z)$ is the generator for the conditioned process of Y with $Y_\tau = z$. The reverse KL-divergence is

$$D_{KL}(\Lambda\|M^*) = \int_0^\tau \mathbb{E}_{\Lambda_{\tau,t}}\left[(\tilde{A}^{\Lambda_{\cdot|\tau}} - \tilde{A}^{M^*})(Y_t, Y_t) + \sum_{x\neq Y_t} \tilde{A}^{\Lambda_{\cdot|\tau}} \log\frac{\tilde{A}^{\Lambda_{\cdot|\tau}}}{\tilde{A}^{M^*}}(Y_t, x)\right] dt,$$

where the $\tilde{A}^{\Lambda_{\cdot|\tau}}$ is the time-reversal generator for the conditioned measure $\Lambda_{\cdot|\tau}$ which is defined as

$$\tilde{A}_t^{\Lambda_{\cdot|\tau}}(Y_t, x) = \mathbb{E}_{\Lambda_{0|t,\tau}}\left[\tilde{A}(Y_t, x; Y_\tau)\middle|Y_t, Y_0\right].$$

Proof. Follow the proof of Proposition A.2. Note that the Markov measure is time-symmetric. □

A.5 Proof of Theorem 2.3

Proof. With iterative projection, we have a sequence of measure $\Lambda^{(n)}$ such that:

$$\begin{aligned}\Lambda^{(2n+1)} &= \Pi_{\mathcal{M}}(\Lambda^{(2n)}),\\ \Lambda^{(2n+2)} &= \Pi_{\mathcal{R}(\mathbb{Q})}(\Lambda^{(2n+1)}),\end{aligned}$$

where $\Lambda^{(0)} \in \mathcal{R}(\mathbb{Q})$. Let $\Pi^{(n)} = \Lambda^{(2n)}$ and $M^{(n)} = \Lambda^{(2n+1)}$. We will omit the superscription $\cdot^{(n)}$ if there is no confusion.

Let $\mathbb{P} \in \mathcal{M}$ be a Markov process generated by a generator $A_t^{\mathbb{P}}(x, y) < \infty$ with initial distribution M_0 with the Kolmogorov forward equation. Assuming that $D_{\mathrm{KL}}(\Pi\|\mathbb{P}) < \infty$, then

$$D_{\mathrm{KL}}(\Pi\|\mathbb{P}) - D_{\mathrm{KL}}(\Pi\|M) = \mathbb{E}_\Pi\left[\log\frac{dM}{d\mathbb{P}}\right] < \infty,$$

which implying the Radon-Nikodym derivative $\frac{dM}{d\mathbb{P}} < \infty$ over the support of Π.

Then we will first show the equality:

$$\mathbb{E}_\Pi\left[\log\frac{dM}{d\mathbb{P}}\right] = \mathbb{E}_M\left[\log\frac{dM}{d\mathbb{P}}\right].$$

As $\Pi_0 = M_0$, it is sufficient to show

$$\mathbb{E}_\Pi\left[\log\frac{dM_{\cdot|0}}{d\mathbb{P}_{\cdot|0}}\right] = \mathbb{E}_M\left[\log\frac{dM_{\cdot|0}}{d\mathbb{P}_{\cdot|0}}\right],$$

where

$$\log\frac{dM_{\cdot|0}}{d\mathbb{P}_{\cdot|0}}(\omega) = \int_0^\tau \log\frac{A_t^M}{A_t^{\mathbb{P}}}(\omega_{t-}, \omega_t)dN_t(\omega) + \int_0^\tau (A_t^M - A_t^{\mathbb{P}})(\omega_t, \omega_t)dt$$

is given from Appendix A.3.1. Since $\Pi_{|0}$ is Markov from Lemma A.3 with generator

$$A_t^{\Pi_{|0}}(x, y) = \mathbb{E}_{\Pi_{\tau|0,t}}\left[A_t(x, y; X_\tau)|X_0, X_t = x\right],$$

which is derived at the Lemma A.5. Then,

$$
\begin{aligned}
&\mathbb{E}_\Pi \left[\int_0^\tau \log \frac{A_t^M}{A_t^{\mathbb{P}}}(X_{t-}, X_t) dN_t \right] \\
&= \mathbb{E}_\Pi \left[\int_0^\tau \sum_{y \neq X_t} A_t^{\Pi_{\cdot|0}} \log \frac{A_t^M}{A_t^{\mathbb{P}}}(X_t, y) dt \right], \\
&= \mathbb{E}_\Pi \left[\int_0^\tau \sum_{y \neq X_t} \mathbb{E}_\Pi \left[A_t(X_t, y; X_\tau) | X_0, X_t \right] \log \frac{A_t^M}{A_t^{\mathbb{P}}}(X_t, y) dt \right], \\
&= \mathbb{E}_\Pi \left[\int_0^\tau \sum_{y \neq X_t} \mathbb{E}_\Pi \left[A_t(X_t, y; X_\tau) | X_t \right] \log \frac{A_t^M}{A_t^{\mathbb{P}}}(X_t, y) dt \right],
\end{aligned}
$$

by the tower property of conditional expectation. From the Lemma A.7 and Proposition A.6, we can replace the $\mathbb{E}_\Pi \left[A_t(X_t, y; X_\tau) | X_t \right]$ as $A_t^M(X_t, y)$,

$$
\begin{aligned}
&\mathbb{E}_\Pi \left[\int_0^\tau \sum_{y \neq X_t} \mathbb{E}_\Pi \left[A_t(X_t, y; X_\tau) | X_t \right] \log \frac{A_t^M}{A_t^{\mathbb{P}}}(X_t, y) dt \right], \\
&= \mathbb{E}_\Pi \left[\int_0^\tau \sum_{y \neq X_t} A_t^M \log \frac{A_t^M}{A_t^{\mathbb{P}}}(X_t, y) dt \right], \\
&= \mathbb{E}_M \left[\int_0^\tau \sum_{y \neq X_t} A_t^M \log \frac{A_t^M}{A_t^{\mathbb{P}}}(X_t, y) dt \right],
\end{aligned}
$$

where the last equation justified since $\Pi_t = M_t$ for all $t \in [0, \tau]$. The other term, $\int_0^\tau (A_t^M - A_t^{\mathbb{P}})(X_t, X_t) dt$, can be treated as the same way, which establishes the equality. From the result, we get $D_{\mathrm{KL}}(\Pi \| \mathbb{P}) = D_{\mathrm{KL}}(\Pi \| M) + D_{\mathrm{KL}}(M \| \mathbb{P})$. The equality is derived for other diffusion processes [19, 20, 6], but in this paper we extended it to continuous Markov chain with discrete state space case.

By letting $\mathbb{P} = \mathbb{P}^{\mathrm{SB}}$, we get

$$
D_{\mathrm{KL}}(\Pi \| \mathbb{P}^{\mathrm{SB}}) \geq D_{\mathrm{KL}}(M \| \mathbb{P}^{\mathrm{SB}}),
$$

the equality holds if and only if $\Pi = M$. By the assumption $D_{\mathrm{KL}}(\Lambda^{(0)} \| \mathbb{P}^{\mathrm{SB}}) < \infty$, the sequence of KL-divergence $\{D_{\mathrm{KL}}(\Lambda^{(n)} \| \mathbb{P}^{\mathrm{SB}})\}_{n \in \mathbb{N}}$ is non-increasing and bounded, which implies the KL-divergence converges. Thus,

$$
\lim_{n \to \infty} D_{\mathrm{KL}}(\Pi^{(n)} \| \mathbb{P}^{\mathrm{SB}}) - D_{\mathrm{KL}}(M^{(n)} \| \mathbb{P}^{\mathrm{SB}}) = \lim_{n \to \infty} D_{\mathrm{KL}}(\Pi^{(n)} \| M^{(n)}) = 0.
$$

We here utilize Aldous' tightness criteria for showing the tightness of $\mathbb{P}^{\mathrm{SB}}$. Since the state space is finite it is sufficient to show the following: For any $\varepsilon > 0$, there exists $\delta > 0$ such that for all stopping time t,

$$
\mathbb{P}^{\mathrm{SB}}(d_{\mathcal{X}}(X_{t+\theta}, X_t) \geq \eta) \leq \varepsilon,
$$

for arbitrary small $\eta > 0$ and $0 < \theta < \delta$ (see Section 16 of [44] Section 3 of [45]). To check the criteria, we will show that the leaving rate of $\mathbb{P}^{\mathrm{SB}}$ is bounded. From the Lemma A.5 and Lemma A.7, we know the analytic form of the transition rate A_t. Since Theorem A.1 ensure that the $\mathbb{P}^{\mathrm{SB}}$ is Markov as well as is in reciprocal class,

$$
\begin{aligned}
A_t^{\mathbb{P}^{\mathrm{SB}}}(x,x) &= \mathbb{E}_{\mathbb{P}^{\mathrm{SB}}}\left[A_t(x,x;X_\tau)|X_0=x_0,X_t=x\right],\\
&= \sum_{X_\tau\in\mathcal{X}}\Bigg[\mathbb{P}^{\mathrm{SB}}(X_\tau|X_t=x,X_0=x_0)A_t(x,x)\\
&\qquad -\frac{\mathbb{P}^{\mathrm{SB}}(X_\tau|X_t=x,X_0)}{P_{t:\tau}(x,X_\tau)}\sum_{u\in\mathcal{X}}A(x,u)P_{t:\tau}(u,X_\tau)\Bigg],\\
&= \sum_{X_\tau\in\mathcal{X}}\Bigg[\mathbb{P}^{\mathrm{SB}}(X_\tau|X_t=x)A_t(x,x)\\
&\qquad -\frac{\mathbb{P}^{\mathrm{SB}}(X_\tau|X_t=x,X_0)}{P_{t:\tau}(x,X_\tau)}\sum_{u\in\mathcal{X}}A(x,u)P_{t:\tau}(u,X_\tau)\Bigg],\\
&= A_t(x,x)-\sum_{X_\tau\in\mathcal{X}}\left[\frac{\mathbb{P}^{\mathrm{SB}}(X_\tau|X_t=x,X_0)}{P_{t:\tau}(x,X_\tau)}\sum_{u\in\mathcal{X}}A(x,u)P_{t:\tau}(u,X_\tau)\right],\\
&= A_t(x,x)-\sum_{X_\tau\in\mathcal{X}}\left[\frac{d\mathbb{P}^{\mathrm{SB}}_{t,\tau}}{d\mathbb{Q}_{t,\tau}}(x,X_\tau)\sum_{u\in\mathcal{X}}A(x,u)P_{t:\tau}(u,X_\tau)\right].
\end{aligned}
$$

The Radon-Nikodym derivative is finite and positive for x, such that $\mathbb{P}^{\mathrm{SB}}(X_t=x)>0$. Because the state space $\mathcal{X}$ is finite, there is finite upper bound u_t for every possible pair (x,x_τ). Thus,

$$
\begin{aligned}
A_t^{\mathbb{P}^{\mathrm{SB}}}(x,x) &\geq A_t(x,x)-\sum_{X_\tau\in\mathcal{X}}\left[u_t\sum_{u\in\mathcal{X}}A(x,u)P_{t:\tau}(u,X_\tau)\right],\\
&= A_t(x,x)-u_t\sum_{u\in\mathcal{X}}A(x,u)\sum_{X_\tau\in\mathcal{X}}P_{t:\tau}(u,X_\tau),\\
&= A_t(x,x)-u_t\sum_{u\in\mathcal{X}}A(x,u),\\
&= A_t(x,x).
\end{aligned}
$$

The leaving rate $c_t^{\mathrm{SB}}(x)$ of $\mathbb{P}^{\mathrm{SB}}$ at (t,x) is bounded by that of Q, $c_t(x)$. Assuming the leaving rate of Q is uniformly bounded by c, c_t^{SB} is uniformly bounded by c. Back to the Aldous' tightness criteria, by choosing $\delta=\varepsilon/c$ we can see that $\mathbb{P}^{\mathrm{SB}}$ is tight.

The sequence $\{\Pi^{(n)}\}_{n\in\mathbb{N}}$ is tight. Since $\mathbb{P}^{\mathrm{SB}}$ is tight, for any $\varepsilon>0$ we can choose a compact and measurable K (under Skorokhod topology and associated Borel σ algebra). For any measurable K,

$$
\begin{aligned}
D_{\mathrm{KL}}(\Pi^{(n)}\|\mathbb{P}^{\mathrm{SB}}) &= \mathbb{E}_{\Pi^{(n)}}\left[-\log\frac{d\mathbb{P}^{\mathrm{SB}}}{d\Pi^{(n)}}|K^c\right]\Pi^{(n)}(K^c)+\mathbb{E}_{\Pi^{(n)}}\left[-\log\frac{d\mathbb{P}^{\mathrm{SB}}}{d\Pi^{(n)}}|K\right]\Pi^{(n)}(K),\\
&\geq -\log\frac{\mathbb{P}^{\mathrm{SB}}(K^c)}{\Pi^{(n)}(K^c)}\Pi^{(n)}(K^c)-\log\frac{\mathbb{P}^{\mathrm{SB}}(K)}{\Pi^{(n)}(K)}\Pi^{(n)}(K),
\end{aligned}
$$

by the Jensen inequality. If $\{\Pi^n\}_{n\in\mathbb{N}}$ is not tight, for each compact K and $\lambda>0$, there is at least one $n\in\mathbb{N}$ where $\Pi^{(n)}(K^c)\geq\lambda,\Pi^{(n)}(K)<1-\lambda$. Thus, for $\varepsilon>0$ there exists $n\in\mathbb{N}$, such that

$$
-\log\frac{\mathbb{P}^{\mathrm{SB}}(K^c)}{\Pi^{(n)}(K^c)}\Pi^{(n)}(K^c)\geq-\log(\varepsilon/\lambda)\lambda,
$$

which implies that the lower bound of $D_{\mathrm{KL}}(\Pi^{(n)}\|\mathbb{P}^{\mathrm{SB}})$ can be arbitrary large. However, the KL-divergence is non-increasing and upper bounded by $D_{\mathrm{KL}}(\Pi^{(0)}\|\mathbb{P}^{\mathrm{SB}})<\infty$, contradiction occurs. Thus, the sequence of measure should be tight. Similarly, we can show the $\{M^{(n)}\}_{n\in\mathbb{N}}$ is tight.

The space (Ω,d_Ω) is a Polish space (Section 12 of [44] or Section 3.5 of [45]), and therefore, by Prokhorov's theorem, the collections of measures ${\Pi^{(n)}}_{n\in\mathbb{N}}$ and ${M^{(n)}}_{n\in\mathbb{N}}$ are relatively compact. Each subsequence of $\{M^{(n)}\}_{n\in\mathbb{N}}$ has a sub-subsequence $\{\Pi^{(i)}\}_{i\geq0}$ weakly converges to $\Pi^{(\infty)}$ as $i\to\infty$. Similarly, there is a sub-subsequence $\{M^{(i)}\}_{i\geq0}$ that converges in law to $M^{(\infty)}$. By the lower semi-continuity of the KL-divergence, $D_{\text{KL}}(\Pi^{(\infty)}\|M^{(\infty)}) \leq \liminf_{i\to\infty}(\Pi^{(i)}\|M^{(i)}) = 0$, which implies the two convergence point is equal to $\mathbb{P}^{(\infty)}$. The resulting measure is Markov and is in $\mathcal{R}(\mathbb{Q})$, because the state space is finite, which deduce $\mathbb{P}^{\text{SB}} = \mathbb{P}^{(\infty)}$. We choose an arbitrary convergent point of sub-subsequence resulting in $\mathbb{P}^{\text{SB}}$, the convergence is ensured (Theorem 2.6 [44]). □

B Graph permutation matching

B.1 Introduction

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as the set of nodes $\mathcal{V} = \{v_i\}$ and the set of edges $\{e_{ij}\}$, where i and j denote the node indices. Under a permutation $\sigma \in S_n$, the structure of the graph $\mathcal{G}$ remains unchanged, as the node and edge sets are invariant to indexing:

$$\sigma(\mathcal{V}) = \{v_{\sigma(i)}\} = \mathcal{V},$$
$$\sigma(\mathcal{E}) = \{e_{\sigma(i)\sigma(j)}\} = \mathcal{E},$$

However, the vectorized representation of a graph, $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is affected by the permutation because $\mathbf{V}$ and $\mathbf{E}$ are treated as ordered sets. Thus, a graph $\mathcal{G}$ can be considered as a set of all permuted version of its vectorized representation:

$$\mathcal{G} = \{\sigma(\mathbf{G}) : \sigma \in S_n\},$$

where the $\mathbf{G}$ is an arbitrarily indexed vectorization of $\mathcal{G}$.

The likelihood of transitioning from an initial graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ to a terminal graph $\mathbf{G}' = (\mathbf{V}', \mathbf{E}')$, denoted $p(\mathbf{G}'|\mathbf{G}) = \mathbb{Q}(X_\tau = \mathbf{G}'|X_0 = \mathbf{G})$, depends on both node and edge correspondences, whereas the likelihood $p(\mathcal{G}'|\mathbf{G})$, which is permutation-invariant, does not. More specifically, for any permutation σ, the likelihood $p(\sigma(\mathbf{G}')|\mathbf{G})$ changes, even though the underlying graph $\mathcal{G}'$ remains unchanged. Calculating the likelihood of the graph itself (as opposed to the graph vector) would require summing the likelihoods of all possible permutations of $\mathbf{G}'$, but this approach is computationally prohibitive due to the factorial number of permutations.

This dependency on graph permutation introduces challenges not only in computing likelihoods but also in constructing reciprocal measures. Ideally, a reciprocal measure over graph domain would account for all permutations of graph vectors. However, as with likelihood computations, the construction of reciprocal measures is highly sensitive on the alignment of graph vectors, making proper handling of these permutations essential for the iterative Markovian fitting (IMF) algorithm.

B.2 Sub-optimal vs. Optimal Permutation

In practice, the key insight is that the likelihood difference of graph vectors between the optimal permutation σ^* and sub-optimal permutations σ is substantial, allowing us to neglect sub-optimal permutations. The design choice of the reference process $\mathbb{Q}$, particularly its signal to noise ratio $\frac{\bar{\alpha}(\tau)}{\bar{\alpha}(0)} \gg 0$ (see Equation (8)), ensures the transitions preserving the initial states are far more probable. As a result, the likelihood for sub-optimal permutations, where node and edge correspondences are mismatched, is expected to be significantly lower than for the optimal permutation. The reason is that even a small mismatch in node alignment between σ^* and σ can cause a large number of edge state mismatches, leading to a dramatic decrease in the total likelihood.

Thus, the likelihood $p(\sigma^*\mathbf{G}'|\mathbf{G}) \gg p(\sigma\mathbf{G}'|\mathbf{G})$ and the contribution of sub-optimal permutation in calculating $p(\mathcal{G}'|\mathcal{G})$ is negligible. It is sufficient that focus solely on the optimal permutation σ^* for the likelihood computation, as the probability of sub-optimal arrangements is effectively zero in practical terms.

The prioritization of optimal permutation could be also utilized in the construction of reciprocal measures. Suppose we have only one graph for initial and terminal, where the one is $\mathcal{G}$ and $\mathcal{G}'$, respectively. Without loss of generality, we assume an even distribution over graph vectors $\mathbf{G} \in \mathcal{G}$, and then consider permutations over $\sigma\mathbf{G}' \in \mathcal{G}'$. Note that $p(\sigma\mathbf{G}'|\mathbf{G}) = p(\sigma'\sigma\mathbf{G}'|\sigma'\mathbf{G})$. According to the static SB solution, the graph vectors $\sigma\mathbf{G}' \in \mathcal{G}'$ are distributed according to the likelihood $p(\sigma\mathbf{G}'|\mathbf{G})$ for each graph vector $\mathbf{G} \in \mathcal{G}$, where the optimal permutation is selected most frequently. In this context, neglecting sub-optimal permutation can be viewed as the static OT solution.

As the original SB problem is formulated with the dynamics over graph "vector" domain, arranging the graph vectors is a part of the SB problem. However, by relying on the optimal permutation, we can reduce complexity of the SB problem aroused by the graph vectors alignment. This strategy ensures the transition from a graph vector to the optimally permuted graph vector, resulting in more reliable convergence of the iterative algorithm.

B.3 Standard formulation of Graph matching problem

In this section we will discuss about the standard formulation of the graph matching problem and how the task of finding the optimal permutation σ^*, which minimizes the negative log-likelihood (NLL) $-\log p(\sigma \mathbf{G}'|\mathbf{G})$, can be framed as a graph matching problem.

Consider two graph vectors $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ and $\mathbf{G}' = (\mathbf{V}', \mathbf{E}')$ along with the cost function $\mathbf{c}^V$ and $\mathbf{c}^E$, which define the cost of mapping nodes and edges, respectively. The binary assignment matrix $\mathbf{X} \in \{0,1\}^{n \times n'}$ represents the matching between nodes, where n and n' denote the number of nodes of $\mathbf{G}$ and $\mathbf{G}'$, respectively. If $v_i \in \mathbf{V}$ matches $v'_a \in \mathbf{V}'$, then $\mathbf{X}_{i,a} = 1$, while all other entries for node v_i are zero.

The total node matching cost is given by $\sum_{\mathbf{X}_{i,a}=1} \mathbf{c}^V(v_i, v'_a)$, where $\mathbf{c}^V(v_i, v'_a)$ denotes the cost of matching from $v_i \in \mathbf{V}$ to $v'_a \in \mathbf{V}'$. Similarly, the total edge matching cost is $\sum_{\mathbf{X}_{i,a}=1, \mathbf{X}_{j,b}=1} \mathbf{c}^E(e_{ij}, e'_{ab})$, where $\mathbf{c}^E(e_{ij}, e'_{ab})$ is the cost of matching from $e_{ij} \in \mathbf{E}$ to $e'_{ab} \in \mathbf{E}'$.

We define the cost matrix $\mathbf{A} \in \mathbb{R}^{nn' \times nn'}$, where the diagonal components $\mathbf{A}_{ia,ia} = \mathbf{c}^V(v_i, v'_a)$ represent node costs and the off-diagonal components $\mathbf{A}_{ia,jb} = \mathbf{c}^E(e_{ij}, e'_{ab})$ represent edge costs. By flattening the assignment matrix to be $\mathbf{x} \in \{0,1\}^{nn'}$, we can write the total cost function f as

$$\begin{aligned} f(\mathbf{x}) &= \sum_{\mathbf{x}_{ia}=1, \mathbf{x}_{jb}=1} \mathbf{c}^E(e_{ij}, e'_{ab}) + \sum_{\mathbf{x}_{ia}=1} \mathbf{c}^V(v_i, v'_a), \\ &= \mathbf{x}^\top \mathbf{A} \mathbf{x}. \end{aligned}$$

Thus, the graph matching problem is find the optimal assignment $\mathbf{x}$ that minimizing cost $f(\mathbf{x})$, which is written in formal:

$$\begin{aligned} \mathbf{x}^* &= \arg\min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A} \mathbf{x}, \\ s.t. &\sum_{i=1}^{n} \mathbf{x}_{ia} \leq 1, \sum_{a=1}^{n'} x_{ia} \leq 1, \end{aligned} \tag{12}$$

where the inequalities become equalities if $n = n'$. This formulation corresponds to the well-known quadratic assignment problem (QAP), which is NP-hard.

The NLL $-\log p(\mathbf{G}'|\mathbf{G})$ can be decomposed by the sum of the NLLs of nodes and edges:

$$-\log p(\mathbf{G}'|\mathbf{G}) = \sum_{\delta_{ia}=1, \delta_{jb}=1} -\log P^E_{0:\tau}(e_{ij}, e'_{ab}) + \sum_{\delta_{ia}=1} -\log P^V_{0:\tau}(v_i, v'_a).$$

By interpreting the NLLs of nodes and edges as cost function $\mathbf{c}^V$ and $\mathbf{c}^E$, respectively, the problem of finding the optimal permutation σ^* can be formulated as a QAP, as in Equation (12).

B.4 Solution method

Exact solution methods for the QAP, such as mixed integer programming (MIP), requires combinatorial optimization, which incurs prohibitive computational costs [46]. Many accelerated algorithms adopt branch-and-bound strategies that utilize bounds of objective functions [47, 48, 49, 50], reducing the exploration space. However, the combinatorial optimization cannot be avoidable in these strategy. Alternatively, continuous relaxation methods [27, 51] solve the QAP in Equation (12) using a continuous vector $\mathbf{x}$, bypassing combinatorial explorations. However, the continuous relaxation yields non-binary vectors, which require a discretization step, potentially introducing errors.

In our work, we compared two continuous relaxation methods: spectral method (SM) and maximum polling method (MPM), followed by the Hungarian algorithm for post-discretization. Both methods iteratively update the continuous version of assignment vector $\mathbf{x}$ as:

$$\mathbf{x}_{k+1} = \frac{\mathbf{x}_k - \mathbf{A} \odot \mathbf{x}_k \varepsilon}{v}, \tag{13}$$

where $\odot$ denotes matrix multiplication for SM and max-pooled matrix multiplication for MPM, and ε denotes step size, and v denotes the normalizer. Specifically, the $\mathbf{A} \odot \mathbf{x}$ for SM is described as

$$(\mathbf{A} \odot \mathbf{x})_{ia} = \mathbf{x}_{ia}\mathbf{A}_{ia:ia} + \sum_{j \in \mathcal{N}_i} \sum_{b \in \mathcal{N}_a} \mathbf{x}_{jb}\mathbf{A}_{ia;jb}.$$

In the MPM case, the operation is defined as

$$(\mathbf{A} \odot \mathbf{x})_{ia} = \mathbf{x}_{ia}\mathbf{A}_{ia:ia} + \sum_{j \in \mathcal{N}_i} \max_{b \in \mathcal{N}_a} \mathbf{x}_{jb}\mathbf{A}_{ia;jb}.$$

In practice, we modified the cost matrix $\mathbf{A}$ slightly for efficiency. To accelerate the process, we neglect all edges corresponding to dummies in $\mathbf{G}'$, which significantly reduces computational cost, making $|\mathcal{N}_a|$ scale linearly with n'. Additionally, a small Gaussian perturbation was applied to $\mathbf{A}$, slightly altering the minima in the continuous vector space. We solved each QAP ten times to compensate the effects of randomness, improving performance with reasonable computational costs (see Appendix B.5).

B.5 Performance of graph matching algorithm

In this section, we compare the performance of the following algorithms under different conditions: (1) SM algorithm, (2) MPM algorithm, and (3) MPM algorithm with randomness. The hyper-parameters $\bar{\alpha}(\tau)/\bar{\alpha}(0) = 0.3$ for the reference process $\mathbb{Q}$.

To evaluate the effectiveness of QAP solvers on molecular graphs, we selected 100 molecules from the ZINC test set. In all experiments, the source molecule was treated as fully connected (with dummy types), while the target molecule retained only its original edges. We performed up to 1,000 iterations of updates according to Equation (13) with a specified tolerance.

To assess the performance of the algorithms, we permuted the molecular graphs and then tested whether the original indices could be recovered, where optimality is achieved with the inverse permutation. For this task, we evaluated the exact matching ratio, which indicates the proportion of cases where the original indices were successfully recovered. Due to the inherent symmetry in molecular graphs, multiple optimal permutations may exist. Therefore, instead of relying solely on exact index recovery, we base the success criterion on the objective function value. If the Negative Log-Likelihood (NLL) error falls below 1e-2, the solution is considered successful.

For different molecule pairs where no optimal permutation is available, which is common in practical scenarios, exact solvers become computationally prohibitive even with relatively small molecular graphs. As such, we evaluated the performance of the QAP solvers by measuring the reduction in NLL between the initial and optimized permutations.

We conducted experiments with various hyperparameters, and the results are illustrated in Table 3. All algorithms successfully found exact matches for the same molecules, but the MPM algorithm performed better than the SM algorithm when applied to different molecule pairs. For all subsequent experiments, we adopted the MPM algorithm with randomness in the cost matrix as the QAP solver.

Table 3: **Comparison of different algorithms based on configuration parameters.** NLL drop represents the improvement in likelihood between randomly paired molecules, while exact match reflects the percentage of permutations successfully recovered after applying a random permutation to the molecules. ↑ and ↓ denote higher and lower values are better, respectively.

Algorithm	Configuration						NLL drop(↑)	Exact matching(↑)
	Pooling	Tolerance	Precision	Max # iterations	Noise coefficient	# trials		
(1) SM	Sum	1e-9	FP64	1000	0	1	2.251	100 %
(2) MPM	Max	1e-5	FP32	1000	0	1	13.256	100 %
(3) MPM + Randomness	Max	1e-5	FP32	1000	1e-6	10	13.256	100 %
(4) Our setting	Max	1e-4	FP32	2500	1e-6	10	14.324	100 %

B.6 Relation to graph edit distance

The graph edit distance (GED) is a widely used and flexible metric for measuring the dissimilarity between two graphs. It is defined as the minimum cost required to transform one graph into another through a sequence of unit operations. Each unit operation can be a removal, substitution, or insertion, and can be applied to either node or edges. The total cost of the transformation is the sum of the costs assigned to these components.

According to [52], finding the minimal-cost transformation between two graphs is equivalent to the QAP problem with an associated cost function. Though both problems are NP-hard, the equivalence is meaningful in that there are numerous approximation algorithms for the QAP problem that operate within polynomial time.

The basic idea for re-formulation into QAP problem involves the introduction of dummy nodes and dummy edges, where removal (insertion) operations could be replaced by substitution into (from) dummies. The cost function of unit operation is defined as $\mathbf{c}^V(v_i, v'_a)$ and $\mathbf{e}^E(e_{ij}, e'_{ab})$ for node replacement and edge replacement, respectively. Let $\alpha = \bar{\alpha}(\tau)/\bar{\alpha}(0)$, the cost replacement is defined as:

$$\mathbf{c}^V(v_i, v'_a) = \begin{cases} -\log \frac{(d^V-1)\alpha+1}{d^V} & \text{if } v_i = v'_a \\ -\log \frac{-\alpha+1}{d^V} & \text{otherwise,} \end{cases}$$

$$\mathbf{c}^E(e_{ij}, e'_{ab}) = \begin{cases} -\log \frac{(d^E-1)\alpha+1}{d^E} & \text{if } e_{ij} = e'_{ab} \\ -\log \frac{-\alpha+1}{d^E} & \text{otherwise,} \end{cases}$$

where d^V and d^E denote the cardinality of $\mathcal{X}^V$ and $\mathcal{X}^E$, respectively. Similar to Equation (12), we can formulate it to a quadratic problem with the objective function f as:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{\mathbf{x}_{ia}=1, \mathbf{x}_{jb}=1} \mathbf{c}^E(e_{ij}, e'_{ab}) + \sum_{\mathbf{x}_{ia}=1} \mathbf{c}^V(v_i, v'_a), \\ &= \sum_{\mathbf{x}_{ia}=1, \mathbf{x}_{jb}=1} -\log P^E_{0:\tau}(e_{ij}, e'_{ab}) + \sum_{\mathbf{x}_{ia}=1} -\log P^V_{0:\tau}(v_i, v'_a), \\ &= p(\sigma \mathbf{G}' | \mathbf{G}), \end{aligned}$$

where, the σ is the graph permutation associated to the assignment vector $\mathbf{x}$.

However, the GED is not same to the NLL. Note that the edit cost functions penalize every operations with same cost $-\log \frac{-\alpha+1}{d}$ except for the identity operation. Though the cost of identity operation is lesser than the others, the optimal transformation would not contains any identity operation. In real, the cost of identity operation does not affect the optimal edit path, implying that GED is not equal but proportion to the NLL, where the difference proportional to the number of the nodes and edges that are equal under the optimal graph matching σ^*. Though the GED is not exactly same to the NLL, the problem is equivalent to the graph matching problem.

This observation provides a clear interpretation of the underlying dynamics of the SB problem, revealing that the associated OT cost is effectively the GED. Therefore, solving the SB problem can be understood as finding the OT plan between graph distributions, where the transport cost is defined by the GED.

C Implementation details

C.1 Parameterization

In this section, we briefly describe the neural network parameterization and the practical training loss. According to Equations (4) and (5), the neural network approximates the generator $A_t^{\Lambda \cdot|0}(x, y)$ and $\tilde{A}_t^{\Lambda \cdot|\tau}(y, x)$, which are formulated as conditional expectation of $A(x, y; z)$ and $\tilde{A}(y, x; z)$, respectively (see Lemma A.5).

The target generator takes the form:

$$A_s(x, y; z) = \begin{cases} A_s(x, y)\frac{P_{s:\tau}(y,z)}{P_{s:\tau}(x,z)}, & \text{if } x \neq y \\ -\sum_{u \neq x} A_s(x, u)\frac{P_{s:\tau}(u,z)}{P_{s:\tau}(x,z)}, & \text{if } x = y \end{cases} \tag{14}$$

, where A_s and $P_{s:\tau}$ are tractable attributes of $\mathbb{Q}$. Let the the transition probability matrix be $P(s, \tau) : [0, \tau]^2 \to \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$, with $P_{s:\tau}(x, y) = \mathbf{e}_x^\top P(s, \tau)\mathbf{e}_y$. Thus, the neural network predicts the distribution of z given X_t, denoted as $z_\theta(t, X_t) \in \mathbb{R}^{|\mathcal{X}|}$, and our parameterization choice is as follows:

$$A_s^{M^\theta}(X_t, y; t, X_t) = A_s(x, y)\frac{\mathbf{e}_y P(s, \tau)z_\theta(t, X_t)}{\mathbf{e}_x P(s, \tau)z_\theta(t, X_t)},$$

for $X_t \neq y$. The time reverse generator is also defined similarly.

Though the loss formulation was defined as continuous manner, we approximate it with the discretizations. Firstly, we will replace $A_t(X_t, X_t; z) - A_t^{M^\theta}(X_t, X_t)$ with as follows:

$$\begin{aligned} A_t(X_t, X_t; z) - A_t^{M^\theta}(X_t, X_t) &\approx \frac{1}{\Delta t}(1 + A_t(X_t, X_t; z)\Delta t) \log \frac{1 + A_t(X_t, X_t; z)\Delta t}{1 + A_t^{M^\theta}(X_t, X_t)\Delta t}, \\ &\approx \frac{1}{\Delta t} P_{t:t+\Delta t}^{\mathbb{Q}_{\cdot|\tau=z}}(X_t, X_t) \log \frac{P_{t:t+\Delta t}^{\mathbb{Q}_{\cdot|\tau=z}}(X_t, X_t)}{P_{t:t+\Delta t}^{M^\theta}(X_t, X_t)}. \end{aligned}$$

Similarly,

$$A_t(x, y; z) \log \frac{A_t(x, y; z)}{A_t^{M^\theta}(x, y)} \approx \frac{1}{\Delta t} P_{t:t+\Delta t}^{\mathbb{Q}_{\cdot|\tau=z}}(x, y) \log \frac{P_{t:t+\Delta t}^{\mathbb{Q}_{\cdot|\tau=z}}(x, y)}{P_{t:t+\Delta t}^{M^\theta}(x, y)}.$$

Thus the discretized loss function corresponding to Equation (4) is:

$$L(\theta) = \sum_{t_i} \frac{1}{\Delta t} \mathbb{E}_{\Lambda_{t_i, \tau}} \left[\sum_y P_{t:t+\Delta t}^{\mathbb{Q}_{\cdot|\tau=z}}(x, y) \log \frac{P_{t:t+\Delta t}^{\mathbb{Q}_{\cdot|\tau=z}}(x, y)}{P_{t:t+\Delta t}^{M^\theta}(x, y)} \right].$$

The time reversed loss can be discretized as same way.

C.2 Neural Network Parameterization and Hyperparameter settings

DDSBM and DBM. Our neural network parameterization is based on Vignac et al. [23], which uses a graph transformer network [53]. In short, it takes as input a noisy graph $\mathbf{G}_t = (\mathbf{V}_t, \mathbf{E}_t)$ and predicts a distribution over the target graphs. Structural and spectral features are also used as inputs to improve the expressivity of neural networks. We refer the reader to Vignac et al. [23] for more details.

For noise scheduling, we employ a slightly different strategy than Vignac et al. [23]. While Vignac et al. [23] uses a cosine schedule for $\bar{\alpha}_t$, we implement a symmetric scheduling of α_t by incorporating α_{min}, as defined below:

$$\alpha(t) = \frac{\partial_t \bar{\alpha}(t)}{\bar{\alpha}(t)} = \cos\left(\frac{t/\tau + s}{1 + s} \cdot \frac{\pi}{2}\right)^2 \cdot (1 - \alpha_{\text{min}}) + \alpha_{\text{min}}, \tag{15}$$

for $0 \leq t \leq \tau/2$, with $\alpha(t) = \alpha(\tau - t)$ for the remaining half of the schedule. For instance, with 100 diffusion steps, $\bar{\alpha}(\tau) \approx 0.95$ when $\alpha_{\text{min}} = 0.999$, and $\bar{\alpha}(\tau) \approx 0.90$ when $\alpha_{\text{min}} = 0.99795$.

We trained DDSBM models with IMF iterations, and DBM models, an one-directional variant of DDSBM with fixed joint molecular pairs, were trained with the same number of gradient updates to ensure consistency. Both DBM and DDSBM reported in this work were trained using four RTX A4000 GPUs. The detailed hyperparameters for training are shown in Table 4.

Graph-to-Graph Translation. For HierG2G and AtomG2G, we used the default settings provided on the official GitHub repository. Both models were trained until maximum epochs by default with a single RTX A4000 GPU.

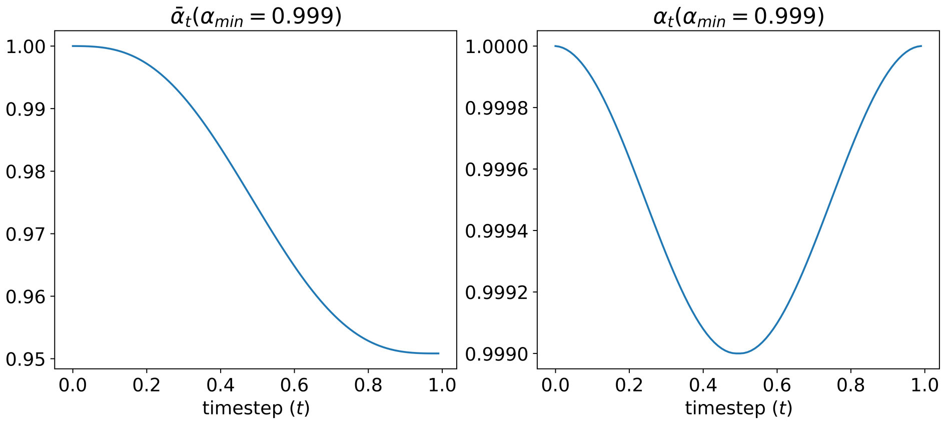


Figure 2: **Plot of $\alpha(t)$ and $\bar{\alpha}(t)$ as functions of timestep t where $\alpha_{\text{min}} = 0.999$**

Table 4: **Training hyperparmeters of DBM and DDSBM.**

Task	Model	diffusion steps	α_{min}	epoch	SB iterations
ZINC250k	DBM	100	0.999	1800	–
ZINC250k	DDSBM	100	0.999	300	6
Polymer	DBM	100	0.999	1250	–
Polymer	DDSBM	100	0.999	250	5

D Supplementary results

D.1 Data processing for ZINC250K dataset experiment

In Section 5.2, we experimented with the standard ZINC250K dataset. In our molecular graph representation, $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, the node vectors $\mathbf{V} = (v^{(i)})_i$ represent the atomic types, and the edge vectors $\mathbf{E} = (e^{(ij)})_{ij}$ represent bond orders. Due to this implementation choice, node features other than the atomic type cannot be represented, causing occasional failures in decoding into molecules. Still, to focus on the SB problem itself, we filtered out molecules whose graph representations are not directly converted into molecules by the RDKit package. Among the node features, formal charge and explicit hydrogen information is crucial in decoding process since, without them, each atom's valency cannot be inferred so that corresponding molecule cannot be uniquely determined. Thus, the following criteria were applied to filter molecules from the ZINC250K dataset: (1) all atoms do not possess a formal charge and (2) all aromatic atoms do not have an explicit hydrogen.

D.2 Examples for generated molecules

To show the difference between molecules generated from DDSBM and others, we visualized some selected examples for the ZINC250K and polymer datasets, respectively (see Figures 3 and 4).

D.3 The effect of initial coupling

We further analyze the molecular optimization task in Section 5.2 with different initial couplings. In that section, we used randomly coupled molecules as the initial coupling for all the models. However, except for DDSBM, all of them assume that suitable molecule pairs have already been identified to be provided. To meet this, we adopted Tanimoto similarity [54] as a pseudo-metric to find similar molecule pairs between the two molecule distributions, denoted as Tanimoto similarity-based coupling. We note that the similarity-based coupling is another optimal transport problem of *maximizing* the sum of pair-wise molecular similarities, where we employed the Hungarian method to obtain a sub-optimal solution.

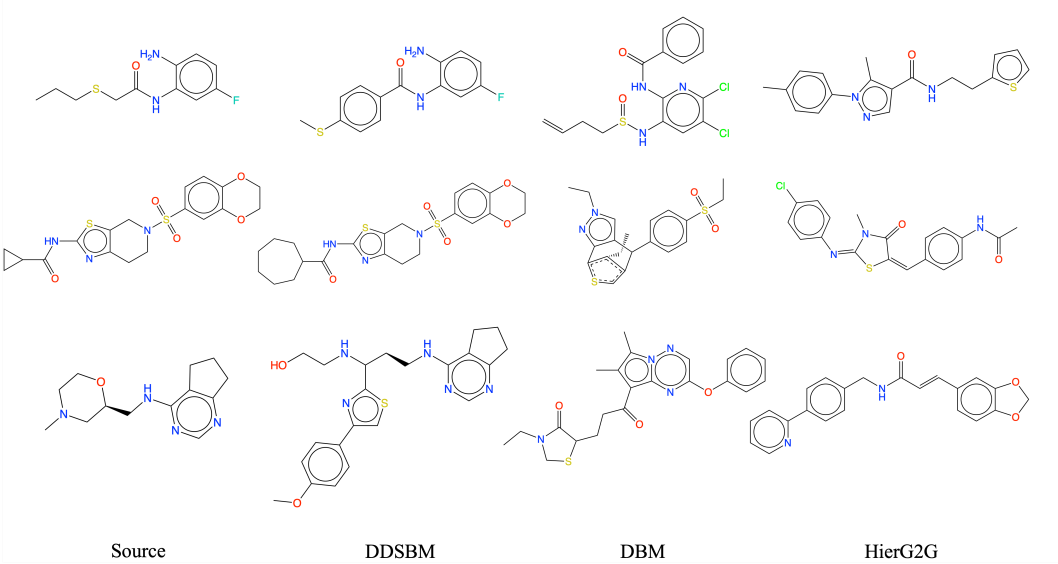


Figure 3: **Visualization of molecules generated by DDSBM, DBM, and HierG2G compared to the source molecule.**

Using the Tanimoto similarity-based coupling, we retrained all models discussed in Section 5.2 and compared their performance. Here, we denote a newly trained DDSBM model as DDSBM-T to deviate it from the model trained on the randomly coupled data. From Table 5, we observe that all models achieved lower NLL values compared to when they were trained with random coupling. The HierG2G model exhibits much lower FCD and NLL values compared to those of random coupling, indicating that previous graph transformation methods can be improved if more optimal pairs are provided as training data.

The distinct feature of DDSBM-T is that all baseline models learn graph transformations between the molecule pairs with high Tanimoto similarity, while DDSBM-T learns graph transformations with minimal cost defined by its reference process $\mathbb{Q}$. It cannot be guaranteed that the distance defined by our reference process is better than the Tanimoto similarity from the perspective of molecular optimization. Essentially, the success of molecular optimization should be measured by how well the target property is adjusted while preserving other key properties. Apparently, Table 5 shows that DDSBM-T outperforms the other baselines in terms of molecular property metrics. This suggests that the graph transformation from DDSBM retains other molecular properties, attaining the goal of the molecule optimization task.

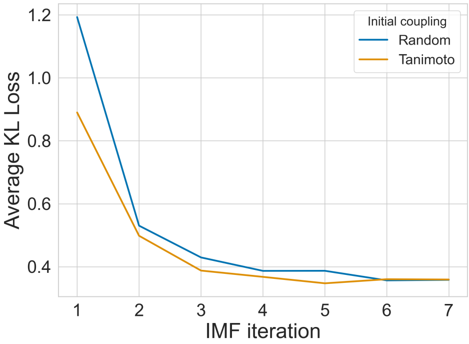


Figure 6: Comparison of $D_{\mathrm{KL}}(\Lambda|M^{\theta})$ across IMF iterations for two types of initial couplings: Random and Tanimoto similarity.

Finally, we analyze the effect of initial coupling on the training process, especially focusing on the approach to convergence. We illustrate the training losses of DDSBM models from two different initial couplings, random and Tanimoto similarity-based, in Figure 6, respectively. DDSBM-T shows consistently lower loss values up to the sixth IMF iteration and reaches convergence at the third IMF iteration.

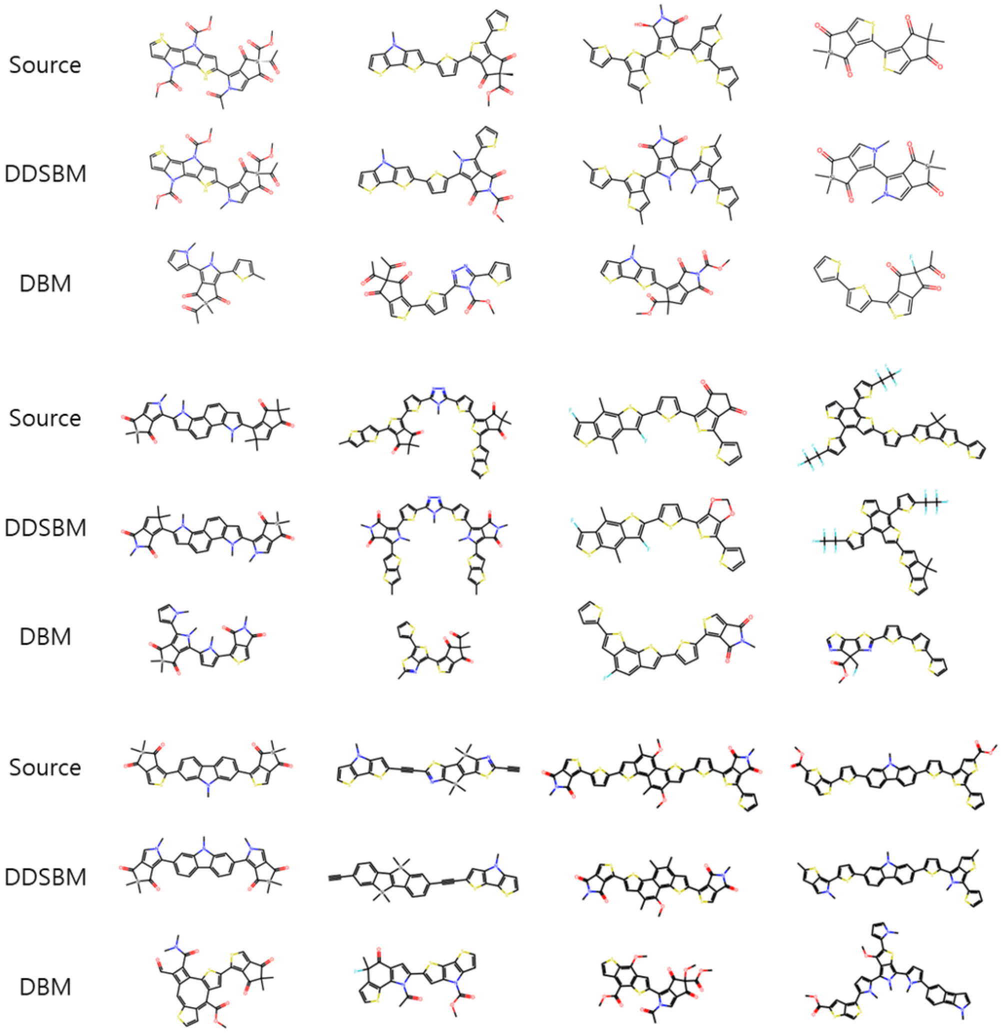


Figure 4: **Visualization of molecules generated by DDSBM and DBM with the source molecule.** The samples generated by DBM and DDSBM were selected from the molecules predicted to have a blue color with GAP values in the range of 2.56–2.75 eV.

Table 5: **Distribution shift performance on ZINC with initial coupling based on the Tanimoto similarity.** As in Table 1, reference refers to metrics from the initial coupling, used as a standard to evaluate each model's graph translation. The experimental setting is the same as described Section 5.2, except for the initial coupling.

Model	Type	Validity(↑)	FCD(↓)	NLL(↓)	LogP W_1(↓)	QED MAD(↓)	SAscore MAD(↓)
Reference[1]	-	-	4.811 / 0.315	245.765	2.011	0.126	0.367
AtomG2G	Latent	**100.0**	5.676	346.499	0.220	0.136	0.738
HierG2G	Latent	**100.0**	1.171	264.072	0.189	0.127	0.446
DBM	Bridge	90.2	**0.749**	220.594	0.141	0.127	0.508
DDSBM-T	Schrödinger Bridge	95.6	0.911	**152.856**	**0.103**	**0.110**	**0.393**

[1] NLL, W_1, and MADs were calculated using random pairs from the test set. Two FCD values are provided: the first compares the initial molecules in the test set with the terminal molecules in the training set, and the second compares the terminal molecules in both sets.

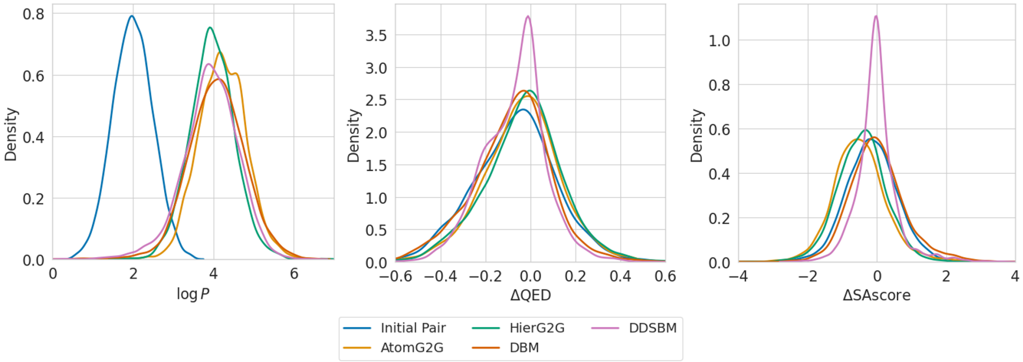


Figure 5: **Density plots comparing the distributions of three molecular properties** (log P, **ΔQED, and ΔSAscore) across four different molecular generation methods: AtomG2G, HierG2G, DBM, and DDSBM.**